

RELEASY

ChatGPT를 이용한
사내 릴리즈 노트 공유 시스템 "Releasy"



TEAM KAKAO99%

학번	이름	학과
201835453	박도영	AI·소프트웨어학부(소프트웨어전공)
201734919	박재석	산업경영공학과
201735979	서강덕	컴퓨터공학과
201835465	서지원	AI·소프트웨어학부(소프트웨어전공)
201835474	안해빈	AI·소프트웨어학부(소프트웨어전공)



목차

0. 보고서 기록 추적
1. 서론
 - 1) 서비스 개요
 - 2) 개발 필요성
2. 사전 조사 및 기존 제품 분석
3. WBS 및 스프린트
4. 요구사항 정의서
5. 기술 스택
6. 코드 컨벤션
7. 와이어프레임 및 스토리보드
8. 서비스 실행 화면
9. 개체-관계 다이어그램(ERD)
10. 데이터 흐름도(DFD)
11. 아키텍처 구조도
12. API 및 인터페이스 명세서
13. QA 보고서
14. 인프라 보고서
15. 성능 테스트 결과 보고서

보고서 기록 추적

버전	날짜	내용
0.0.1.	2023. 7. 3.	보고서 작성 시작
0.1.0.	2023. 7. 27.	프로토타입 보고서 작성 완료
1.0.0	2023. 8. 16	1차 릴리즈 보고서 작성 완료

1. 서론

1) 서비스 개요

“Releasy”는 릴리즈 노트 공유를 쉽고 빠르게 작성하고 팀원들과 공유할 수 있으며 프로젝트 중 발생하는 이슈에 대해서 빠르게 대처할 수 있는 사내 릴리즈 노트 공유 웹 서비스이다. 사내 릴리즈 노트 공유 시스템인 만큼 그룹을 생성하고 각 그룹 내에서 프로젝트를 진행할 수 있으며, 프로젝트마다 그룹 내에서 참여하는 멤버가 상이할 수 있다. “Releasy”는 ‘ChatGPT를 이용한 이슈 별 중요도 및 우선순위 추천 서비스’와 ‘드래그 앤 드롭(Drag and Drop)이 가능한 칸반 보드(Kanban Board)를 통한 이슈 관리 기능’을 강점으로 내세운다.

‘ChatGPT를 이용한 이슈 별 중요도 및 우선순위 추천 서비스’는 현재 프로젝트 중 발생한 이슈들의 목록에서 PM이 ChatGPT를 통해 중요도와 우선순위를 추천받고, 이를 적용함으로써 프로젝트의 진행 순서를 명확하고 빠르게 결정하여 이슈와 업무를 분배할 수 있도록 돕는 서비스이다. PM은 직접 중요도를 지정할 수도 있지만, 본 서비스를 이용함으로써 보다 객관적이고 신속한 의사결정이 가능할 것으로 기대된다. 결정된 중요도는 프로젝트 대시보드와 이슈 페이지에 표시되며, 수치와 색상으로 구분한다.

‘드래그 앤 드롭(Drag and Drop)이 가능한 칸반 보드(Kanban Board)를 통한 이슈 관리 기능’은 프로젝트 중 생성된 이슈 또는 업무의 단계와 상태를 시각적으로 명확히 파악할 수 있도록 돕는다. 칸반(Kanban)은 반복적인 프로세스의 단계를 명확히 나누는 프로젝트 관리 방식으로, 발행한 이슈와 업무 목록을 각각 Backlog(진행 전), In Progress(진행 중), Done(완료)의 3가지로 구분하고 각 항목을 드래그 앤 드롭으로 이동시켜 상태를 변경할 수 있으며 이전에서 언급한 중요도와 함께 정보가 함께 표시된다.

또한, 본 목적은 사내 릴리즈 노트의 공유이기 때문에 릴리즈 노트의 작성 및 수정을 직관적이고 간단하게 나타내었으며, 해당 릴리즈 노트와 관련된 이슈들을 직접 추가 및 삭제할 수 있도록 하였다. 실제 해당 버전의 릴리즈 전에 상태를 ‘릴리즈 전’으로 지정해 두고 미리 릴리즈 노트를 작성하여 저장해 둘 수 있으며, 쉽게 상태 및 종속된 이슈, 내용을 변경할 수 있다,

2) 개발 필요성

프로젝트 매니저(이하 PM)은 프로젝트 진행 중에 모든 일정과 진행 상황을 통제 및 조율해야 한다. PM이 신경 써야 할 부분이 굉장히 많기 때문에 조금이라도 부담을 덜기 위해서 “Releasy”는 이슈의 우선순위를 결정과 팀원에게 할당 시에 ChatGPT를 통한 추천을 받을 수 있도록 기획하였다. 또한, 이러한 진행 상황을 팀 구성원 모두가 쉽게 파악할 수 있어야 하기 때문에 칸반(Kanban) 보드를 통해 쉽게 파악할 수 있도록 했으며, 드래그 앤 드롭 기능을 통해 자신이 배정받은 이슈의 현재 상태를 쉽고 빠르게 변경하고 팀원들에게 공유할 수 있도록 기획하였다.

2. 사전 조사 및 기존 제품 분석

릴리즈 노트(release note)는 소프트웨어 또는 서비스가 출시 또는 업데이트 될 때마다 소프트웨어 제품과 함께 배포되는 문서들을 말하며, 제품이 개발 중이거나 테스트 상태(베타 릴리스 등)일 때 추가되기도 한다. 고객이 이미 사용 중인 제품의 경우 릴리스 노트는 업데이트가 출시될 때 고객에게 전달된다. 릴리즈 노트는 종이 매뉴얼, 모바일/웹 애플리케이션, 개발자 사이트 등 다양한 소스에 게시될 수 있다.

- 릴리즈 노트를 통해 소프트웨어의 버전과 릴리즈 정보를 체계적으로 관리 할 수 있다.
- 소프트웨어의 전체 기능, 서비스의 내용, 개선 사항, 버전 정보 등을 사용자와 공유하는 것이다.
- 업데이트 정보를 한데 모아 기록함으로써, 내가 작업하지 않은 업데이트 사항도 빠르게 파악 가능하다.
- 릴리즈 노트를 보면 누구나 제품의 발자취를 한눈에 파악 가능하다.
- 우리 회사의 제품 혹은 서비스가 지속적으로 개선되고 최신의 상태를 유지하고 있음을 안내한다.
- 사용자의 신고와 의견을 반영하고 있다는 인상을 주어 사용자의 적극적인 피드백을 유도한다.

1) Notion Projects

문서, 지식, AI와 통합된 프로젝트 관리로 업무를 빠르고 명확하게 만들 수 있도록 지원한다.

- AI 자동 채우기: 요약본, 액션 아이템, 프로젝트 업데이트 등을 자동으로 생성한다.
- 트래킹이 필요한 작업과 프로젝트의 업데이트 알림을 제공한다.
- 스프린트 플래닝: 베타 버전에서 사전 구성된 스프린트 워크플로를 사용할 수 있다.
- 이슈마다 고유 ID 부여: 어느 톨의 이슈와 작업이든 간편하게 연결하고 커뮤니케이션할 수 있다.
- GitHub, Slack 등 API 통합으로 모든 작업 연결: GitHub PR 상태 변경 시 Notion 작업을 자동 업데이트 하는 등의 여러 작업을 지원한다.
- Figma와 Google Drive 파일을 작업에 연결할 수 있도록 지원할 예정이다.
- 변경 사항이 있을 때 수행할 작업을 미리 지정할 수 있도록 할 예정이다.

2) Github Release

GitHub에서는 소프트웨어의 소스코드 뿐만 아니라 소프트웨어를 빌드한 결과물을 배포할 수 있는 기능까지 제공하고 있다. '릴리즈(Release)' 라는 기능이다.

- Master branch의 태그와 함께 릴리즈 제목, 릴리즈 노트 및 소스코드를 빌드한 바이너리를 첨부할 수 있다.
- 릴리즈 정보 자동 생성: Repository의 코드와 파일들을 바탕으로 릴리즈 정보를 수동으로 작성하는 작업을 자동화하는 기능을 제공한다.
- GitHub-release: GitHub에 공개되어 있는 'aktau/github-release'는 CLI 환경을 이용해 깃허브 릴리즈를 쉽게 할 수 있도록 해준다. CLI 환경에서 사용 가능하므로 자동화된 스크립트를 만들기도 좋다. 'github-release' 명령을 통해서 깃허브 릴리즈 페이지에 파일을 업로드하고, 릴리즈 노트를 수정할 수도 있다.

3) Slack

Slack은 팀과 업무를 하나로 이어주는 단일 플랫폼이다. 클라우드 기반의 팀 협업 도구로, 실시간 메시지, 파일 공유, 작업 관리 등 많은 기능들을 제공하여 사람들이 한 곳에서 통합된 하나의 팀을 운영하여 조직이 커뮤니케이션하는 방법을 개선할 수 있다. 업무에 관련된 모든 커뮤니케이션이 한 곳으로 집중되어 빠른 업무 확인 및 의사 결정이 가능하여 이메일을 대체할 수 있는 커뮤니케이션 도구이다. 다음과 같은 장점들을 가지고 있다.

- 실시간 커뮤니케이션: 채팅 기능을 통해 실시간으로 팀원들과 대화할 수 있다.
- 팀 협업과 프로젝트 관리: 채널을 생성하여 특정 팀, 프로젝트 또는 주제 별로 그룹을 형성할 수 있다. 그룹 내에서 여러 팀원들이 동시에 파일 공유와 협업 기능을 활용하여 문서 공유, 편집, 리뷰 등을 효율적으로 수행할 수 있다.
- 이력 및 검색 기능: 과거의 대화 내용을 쉽게 찾아보고, 필요한 정보나 파일을 빠르게 검색할 수 있다.
- 화상회의 기능, 허들(Huddle): 채널의 인원들과 음성 또는 화상으로 즉시 연결할 수 있다. 화상통화 중 공유한 모든 링크, 문서, 메시지는 자동으로 저장되기 때문에 허들이 종료되어도 쉽게 다시 참조할 수 있다.
- 워크플로: 코드 없이 자동화된 작업을 만들게 도와준다. 누군가 채널에 들어왔을 때나, 채널에 속한 누군가가 특정 반응을 추가했을 때 등에 설정해 놓은 단계별 프로세스가 자동으로 실행되는 기능이다. 노 코드(No-Code) 방식이기 때문에 누구나 쉽게 워크플로를 생성하고 사용할 수 있다..
- 다양한 앱과 통합: 2600개 이상의 앱과의 통합을 지원하여 업무 생산성을 높일 수 있다.

3. WBS 및 스프린트 (7/27 기준)

Aa 항목	📄 업무 내용	📄 산출물	👤 작업자	📅 시작일	📅 종료일	📊 작업 일수	🟢 Status	📈 진행 정도 (%)
기존 제품 분석		기존 제품 분석 보고서	👤 안해빈	2023년 6월 28일	2023년 6월 30일	2	Done	100 <div></div>
요구사항 정의		요구사항 정의서	👤 지원 서	2023년 7월 3일	2023년 7월 7일	7	Done	100 <div></div>
스토리보드 작성		스토리보드	👤 박도영 👤 박재석 👤 지원 서 👤 안해빈 👤 서강덕	2023년 7월 3일	2023년 7월 7일	2	Done	100 <div></div>
컴포넌트 구성		컴포넌트 구성 다이어그램					Done	100 <div></div>
ChatGPT 다이어그램 작성		GPT 다이어그램	👤 박도영	2023년 7월 4일	2023년 7월 4일	1	Done	100 <div></div>
DB 설계		ERD	👤 박도영	2023년 7월 6일	2023년 7월 7일	2		
아키텍처 설계		아키텍처 설계 다이어그램		2023년 7월 6일	2023년 7월 7일	2	Done	100 <div></div>
깃 전략 수립		Git Flow 보고서		2023년 7월 7일	2023년 7월 7일	1	Done	100 <div></div>
코드 컨벤션 수립		코드 컨벤션 보고서	👤 서강덕	2023년 7월 7일	2023년 7월 7일	1	Done	100 <div></div>
엔티페이스 정의		엔티페이스 정의서	👤 박재석	2023년 7월 10일	2023년 7월 12일	3	Done	100 <div></div>
기능 개발 - 1 [프론트엔드]	멤버 관리	UI/UX	👤 서강덕	2023년 7월 12일	2023년 7월 19일	5	Done	100 <div></div>
	필러즈노트	UI/UX	👤 안해빈	2023년 7월 12일	2023년 7월 19일	5	Done	100 <div></div>
	메모 관리	UI/UX	👤 박재석	2023년 7월 12일	2023년 7월 19일	5	Done	100 <div></div>
	프로젝트 관리	UI/UX	👤 지원 서	2023년 7월 12일	2023년 7월 19일	5	Done	100 <div></div>
	이슈 관리	UI/UX	👤 박도영	2023년 7월 12일	2023년 7월 19일	5	Done	100 <div></div>
기능 개발 - 2 [백엔드]	멤버 관리	API	👤 서강덕	2023년 7월 19일	2023년 7월 26일	5	Done	100 <div></div>
	필러즈노트	API	👤 안해빈	2023년 7월 19일	2023년 7월 26일	5	Done	100 <div></div>
	메모 관리	API	👤 박재석	2023년 7월 19일	2023년 7월 26일	5	Done	100 <div></div>
	프로젝트 관리	API	👤 지원 서	2023년 7월 19일	2023년 7월 26일	5	Done	100 <div></div>
	이슈 관리	API	👤 박도영	2023년 7월 19일	2023년 7월 26일	5	Done	100 <div></div>
프로토타입 마무리		프론트엔드 & 백엔드 통합	👤 박도영 👤 박재석 👤 서강덕 👤 안해빈 👤 지원 서	2023년 7월 26일	2023년 7월 28일	5	Done	100 <div></div>
발표 준비		발표 준비 및 프로토타입 마무리	👤 박도영 👤 안해빈 👤 박재석 👤 서강덕 👤 지원 서	2023년 7월 24일	2023년 7월 27일	2	Done	100 <div></div>
중간 발표		중간 발표	👤 박도영	2023년 7월 28일	2023년 7월 28일	1	In progress	
1차 QA		1차 QA 보고서		2023년 7월 31일	2023년 8월 1일	2	Not started	
1차 수정 개발		1차 수정 개발 보고서		2023년 8월 1일	2023년 8월 4일	4	Not started	
기능 개발 - 3		ChatGPT - 종요도		2023년 8월 1일	2023년 8월 7일	7	Not started	
		RedditMQ - push 알림		2023년 8월 1일	2023년 8월 7일	7	Not started	
		Elastic Search - 검색		2023년 8월 1일	2023년 8월 7일	7	Not started	
클라우드 환경 구축	KIC구축	(기간 변동 가능)		2023년 8월 2일	2023년 8월 4일	3	Not started	
CI/CD 적용		CI/CD 적용		2023년 8월 3일	2023년 8월 4일	2		
기능 개발 - 4		Level Up + 잔디밭		2023년 8월 8일	2023년 8월 14일	7	Not started	
		로그인 현황 조회		2023년 8월 8일	2023년 8월 14일	7		
2차 QA		2차 QA 보고서		2023년 8월 9일	2023년 8월 11일	3	Not started	
2차 수정 개발		2차 수정 개발		2023년 8월 14일	2023년 8월 16일	3	Not started	
최종 배포		최종 배포		2023년 8월 16일	2023년 8월 16일	1	Not started	
회고		회고		2023년 8월 17일	2023년 8월 17일	1	Not started	
발표 준비		발표 준비		2023년 8월 14일	2023년 8월 17일	4	Not started	
최종 발표		최종 발표		2023년 8월 18일	2023년 8월 18일	1	Not started	

4. 요구사항 정의서

REQ-ID	페이지	서비스 구분	사용자 구분	요구사항	내용	구분
REQ-1	모든 페이지	성능	모든 사용자	시스템 사용자 수 처리	시스템은 성능 저하 없이 필요한 수의 사용자를 처리할 수 있어야 한다.	비기능
REQ-2	모든 페이지	보안	모든 사용자	시스템 보안	시스템은 무단 액세스로부터 안전해야 한다.	비기능
REQ-3	모든 페이지	성능	모든 사용자	페이지 응답 시간	웹사이트의 응답시간은 3 초 이내여야 한다.	비기능
REQ-4	모든 페이지	접근성	모든 사용자	사용 친화적 화면	웹사이트의 모든 화면은 명확하고 사용 친화적이며 누구나 기능을 이해하고 사용할 수 있어야 한다.	비기능
REQ-5	홈	메인 랜딩	모든 사용자	홈페이지 제공	사진, 설명, 로그인 버튼, 회원가입 버튼이 정상적으로 노출되어야 한다.	기능
REQ-6	로그인	로그인/보안	모든 사용자	소셜 로그인 지원	카카오, 구글, 네이버의 소셜 미디어 계정으로 로그인 할 수 있어야 한다.	기능
REQ-7	로그인	로그인/보안	모든 사용자	이메일 로그인 지원	이메일과 비밀번호를 이용하여 로그인할 수 있어야 한다.	기능
REQ-8	로그인	로그인/보안	모든 사용자	로딩 속도	로그인 시 3 초 이내에 완료되어야 한다.	비기능
REQ-9	로그인	로그인/보안	모든 사용자	입력 횟수	비밀번호를 5 번 이상 틀린 경우 일정 시간 동안 로그인 시도를 제한해야 한다.	기능
REQ-10	로그인	로그인/검증	모든 사용자	비밀번호 검사	사용자가 입력한 비밀번호와 암호화 된 비밀번호가 일치하는지 확인해야 한다.	기능
REQ-11	회원가입	회원가입	모든 사용자	유효성 검사	회원가입 양식에 유효성 검사가 적용되어야 한다.	기능
REQ-12	회원가입	회원가입	모든 사용자	중복 방지	동일한 email 정보로 가입하는 걸 방지해야 한다.	기능

REQ-13	회원가입	비밀번호	모든 사용자	암호화	회원 가입 시 입력하는 비밀번호는 암호화 하여 안전하게 보관해야 한다.	비기능
REQ-14	회원가입	비밀번호	모든 사용자	암호화	회원 가입 시 입력하는 비밀번호가 동일해도 암호화 결과는 달라야 한다.	비기능
REQ-15	회원가입	그룹 생성	모든 사용자	유효성 검사	생성하고자 하는 그룹명의 중복 여부를 검사해야 한다.	기능
REQ-16	회원가입	그룹 참여	모든 사용자	유효성 검사	입력한 그룹 코드가 존재하는지 검사해야 한다.	기능
REQ-17	로그인/회원가입	관리자 도구	시스템 관리자	로그인 설정 및 확인	이메일 및 소셜 로그인에 대해 설정하고 정보를 확인할 수 있어야 한다.	기능
REQ-18	로그인/회원가입	입력	모든 사용자	유효성 검사	빈 입력 란이 존재하거나, 입력 형식에 맞지 않는 란이 존재하면 양식을 제출할 수 없도록 해야 한다.	기능
REQ-19	로그인/회원가입	JWT 토큰	모든 사용자	JWT 토큰 저장 방식	JWT 토큰은 쿠키로 저장되어 사용된다.	비기능
REQ-20	프로젝트 관리 페이지	프로젝트 관리	그룹 멤버	프로젝트 및 그룹 구분	그룹 참여자는 참여 중인 프로젝트와 그룹의 다른 프로젝트를 볼 수 있다.	기능
REQ-21	프로젝트 관리	페이지 이동	그룹 멤버	프로젝트 목록 확인 및 이동	그룹 참여자는 참여 중인 프로젝트의 목록을 확인하고 프로젝트 페이지로 이동할 수 있다.	기능
REQ-22	프로젝트 관리 페이지	프로젝트 조회	그룹 멤버	릴리즈 노트 확인	-그룹 참여자는 그룹의 다른 프로젝트들의 릴리즈 노트를 조회할 수 있다. -프로젝트에 속해있지 않은 그룹의 프로젝트 대시보드로는 이동할 수 없다.	기능
REQ-23	프로젝트 관리	프로젝트 수정	GM, PM	프로젝트 수정 및 삭제 기능	GM 또는 각 프로젝트의 PM 은 프로젝트의 이름, 설명, 상태를 수정할 수 있다.	기능
REQ-24	프로젝트 관리	프로젝트 삭제	GM, PM	프로젝트 삭제 기능	GM 또는 각 프로젝트의 PM 은 프로젝트를 삭제할 수 있다.	기능

REQ-25	프로젝트 생성	프로젝트 생성	그룹 멤버	프로젝트 생성 기능	그룹 멤버들은 프로젝트 생성 시 프로젝트 이름, 상태, 설명을 작성해서 프로젝트를 생성할 수 있다.	기능
REQ-26	내 정보	정보	그룹 멤버	정보 조회 기능	그룹 멤버는 나에 대한 정보(이름, 핸드폰번호, 이메일, 직책)와 다른 그룹 멤버들의 정보(이름, 직책, 입사일, 이메일)을 확인할 수 있다.	기능
REQ-27	내 정보	정보	그룹 멤버	정보 검색 기능	그룹 멤버는 이름을 통해서 검색이 가능하다.	기능
REQ-28	내 정보	그룹 멤버	GM	그룹 멤버 삭제	그룹 매니저는 현재 내 그룹의 멤버를 삭제할 수 있다.	기능
REQ-29	사이드바	프로젝트 페이지 사이드바	프로젝트 멤버	프로젝트 하위 페이지로의 이동	해당 프로젝트의 하위 기능 페이지로 선택하여 이동할 수 있도록 한다.	기능
REQ-30	페이지 설정	프로젝트 페이지 설정	프로젝트 멤버	프로젝트 페이지 UI 수정	<ul style="list-style-type: none"> -UI 설정 메뉴에서 색을 눌러서 메뉴 바 색상을 변경할 수 있다. - 메뉴바 타입을 어둡게, 투명하게, 밝게 중에서 선택해서 변경할 수 있다. -메뉴 바 고정을 토글로 설정할 수 있다. -토글을 활용해서 다크모드로 전환할 수 있다. 	기능
REQ-31	대시보드	프로젝트 대시보드	프로젝트 멤버	각 프로젝트 요약 정보 확인	해당 프로젝트의 요약된 정보를 그래프와 표로 한 눈에 확인할 수 있어야 한다. (월별 배포한 릴리즈 노트 수, 월별 해결한 작업 수, 현재 처리 중인 이슈 목록, 시간 순 알림)	기능
REQ-32	대시보드	시각화(그래프)	프로젝트 멤버	그래프 표현	월별 배포한 릴리즈 노트 수, 월별 해결한 작업 수를 그래프로 나타낼 때 대시보드 페이지가 표시되고 2 초 안에 그래프 애니메이션이 완료되고 표시되어야 한다.	비기능
REQ-33	대시보드	이슈 목록	프로젝트 멤버	이슈 정렬	중요도 순, 기한 순으로 각각 정렬할 수 있어야 하며 나에게 배정된 이슈만 확인할 수도 있어야 한다.	기능

REQ-34	대시보드	이슈 목록	프로젝트 멤버	중요도	중요도는 게이지 바 형태로 표시되며, 중요도(정수)에 따라 색을 다르게 해서 구분해야 한다.	기능
REQ-35	대시보드	이슈 중요도 지정	PM	수동 또는 GPT 를 이용한 이슈 중요도 지정	- PM 은 직접 이슈에 대한 각각, 또는 전체의 중요도를 지정할 수 있어야 한다. - GPT 를 이용한 중요도 지정 기능을 이용하면 자동으로 판단해서 중요도를 지정해 주며, 적용 또는 취소 버튼을 눌러 진행할 수 있다.	기능
REQ-36	대시보드	알림	프로젝트 멤버	주요 변경정보 알림 기능	자신이 속한 프로젝트의 주요 변경 사항(이슈 생성, 이슈 담당자 지정, 이슈 수정, 릴리즈노트 생성 등)이 생길 때 그 정보를 기록해두고 전 프로젝트 멤버가 그 내용을 알림으로 확인할 수 있어야 한다.	기능
REQ-37	멤버정보	멤버정보	프로젝트 멤버	멤버 정보 조회	멤버 이름, 역할(직급), 상태(온/오프라인),입사일을 표 형식으로 확인 할 수 있어야 한다. 이름을 클릭할 경우 해당 멤버의 프로파일 페이지로 이동할 수 있어야 한다.	기능
REQ-38	멤버정보	멤버정보	프로젝트 멤버	멤버 정보 필터링 및 색검색	멤버 직급, 직책으로 필터링 할 수 있어야하며, 이름으로 검색하여 멤버를 조회할 수 있어야 한다.	기능
REQ-39	멤버들의 프로젝트	멤버들의 프로젝트	프로젝트 멤버	프로젝트 멤버의 프로젝트 조회	화면 좌측에 멤버 목록이 표시 되어야 하며, 멤버를 클릭할 경우 화면 우측에서 해당 멤버의 프로젝트 이름 ,상태(진행중,완료됨,취소됨), 생성일을 조회할 수 있어야 한다.	기능
REQ-40	멤버들의 프로젝트	멤버들의 프로젝트	프로젝트 멤버	프로젝트 멤버 필터링 및 검색	멤버 직급, 직책으로 필터링 할 수 있어야하며, 이름으로 검색하여 멤버를 조회하여야 한다.	기능
REQ-41	이슈검색	이슈 조회	프로젝트 멤버	검색	검색창에서 이슈 제목으로 이슈를 검색할 수 있어야 한다.	기능

REQ-42	이슈검색	이슈 조회	프로젝트 멤버	필터링	<ul style="list-style-type: none"> - 이슈의 상태(백로그, 진행중, 완료),이슈의 타임(bug, story,task) 로 구분해서 이슈를 필터링할 수 있어야 한다. - 이슈의 담당자로 이슈를 필터링해서 보여줘야 한다. 	기능
REQ-43	이슈검색	이슈 조회	프로젝트 멤버	성능	빠른 응답 시간과 처리량을 제공할 수 있어야능 한다. 이슈를 신속하게 제출하고 조회 할 수 있어야 한다.	비기능
REQ-44	이슈검색	이슈 조회	프로젝트 멤버	사용자 경험	- 사용자가 쉽게 이해하고 사용할 수 있도록 직관적이고 편리한 인터페이스를 제공해야 한다.	비기능
REQ-45	이슈검색	세부내용 조회	프로젝트 멤버	이슈 세부 내용 조회	<ul style="list-style-type: none"> - 이슈 타입, 이슈 제목, 설명이 보여진다. - 하부의 활동란에는 댓글과 기록이 보여진다. - 추가된 댓글은 기록에서 보여진다. - 이슈의 담당자와 보고자가 보여진다. 	기능
REQ-46	이슈검색	세부내용 수정	PM, 이슈 보고자	이슈 세부 내용 수정	<ul style="list-style-type: none"> - 사진 첨부는 1 장까지 가능하다. - 댓글을 달 수 있다. - 하위 이슈 추가버튼을 눌러서 하위 이슈를 추가할 수 있다. 추가한 하위 이슈도 모든 이슈 목록에 포함된다. - 다른 이슈와 연결할 수 있다. 이슈 연결 버튼을 누르면 이전 이슈의 목록이 보여진다. 	기능
REQ-47	이슈검색	세부내용 수정	PM, 이슈 보고자	이슈 삭제	이슈를 삭제하면 이슈 목록에서 사라진다.	기능
REQ-48	이슈관리	전체 이슈 목록	프로젝트 멤버	전체 이슈 조회	<ul style="list-style-type: none"> - 프로젝트 내의 모든 이슈가 상태(백로그/진행중/완료)에 따라 구분되어 보여진다. - 이슈 제목으로 검색할 수 있다. - 상태, 담당자로 필터링해서 볼 수 있다. 	기능

REQ-49	이슈관리	이슈 칸반 보드	프로젝트 멤버	이슈 드롭 다운	<p>-프로젝트의 이슈들은 드롭다운 기능을 활용해서 상태(백로그/진행중/완료)를 변경할 수 있다.</p> <p>-칸반보드에서 이슈상태를 변경할 때마다 DB 에 바뀐 결과가 적용된다.</p> <p>- 이슈관리 페이지는 2 초마다 칸반보드를 업데이트해서 결과가 바뀌면 적용되게 한다.</p>	기능
REQ-50	이슈관리	이슈 생성	프로젝트 멤버	이슈 생성	<p>- 새로운 이슈를 생성할 수 있다.</p> <p>- 이슈 생성 버튼을 누르면 생성을 위한 모달창이 보여진다.</p> <p>- 사진 첨부는 1 장만 가능하다.</p> <p>- 하위 이슈를 추가할 수 있다.</p> <p>- 설명과 댓글, 활동 표시를 할 수 있다.</p> <p>- 상태, 담당자, 분류, 수정 버전, 보고자를 지정할 수 있다.</p> <p>- 시간은 현재 시간으로 저장된다.</p>	기능
REQ-51	이슈 관리	세부내용 수정	PM, 이슈 보고자	첨부파일 용량	첨부파일의 크기는 1MB 보다 작아야한다.	비기능
REQ-52	댓글검색	댓글 조회	프로젝트 멤버	댓글 조회	<p>-특정 이슈에 작성된 모든 댓글을 조회할 수 있어야 한다.</p> <p>-댓글이 많은 경우, 페이지네이션을 지원하여 사용자가 원하는 범위의 댓글을 확인할 수 있어야 한다.</p> <p>- 특정 기준(작성자, 날짜) 에 따라 댓글을 필터링 할링 수 있어야 한다.능</p>	기능
REQ-53	댓글검색	댓글 조회	프로젝트 멤버	신속한 응답시간	댓글은 실시간으로 동작하여 사용자들이 빠르게 댓글을 작성하고, 이에 대한 응답을 받을 수 있어야 한다.	
REQ-54	댓글생성	댓글 생성	프로젝트 멤버	댓글 생성	사용자가 댓글을 작성할 수 있어야 한다.	기능

REQ-55	댓글 수정	댓글 수정	프로젝트 멤버	댓글 수정	<p>-사용자는 자신이 작성한 댓글을 수정할 수 있어야 한다. 수정된 댓글은 즉시 반영되어야 한다.</p> <p>- 댓글 수정이 완료된 경우, 수정 이력이 표시되어야 한다.</p> <p>- 댓글 수정에 대하여 해당 댓글 작성자에 한하여 가능하게 해야한다.</p> <p>댓글 수정 버튼은 댓글 작성자와 관리자에 한하여 볼 수 있어야 한다.</p>	기능
REQ-56	댓글 삭제	댓글 삭제	프로젝트 멤버	댓글 삭제	<p>사용자는 자신이 작성한 댓글을 작성 할 수 있어야 한다 삭제된 댓글을 즉시 사라져야 한다.</p> <p>댓글 삭제에 대하여 댓글 작성자와 관리자에 한하여 가능하게 해야 한다.</p> <p>댓글 삭제 버튼은 댓글 작성자와 관리자에 한하여 볼 수 있어야 한다.</p>	기능
REQ-57	프로필	내 정보 관리	모든 사용자	내 정보 조회	이름, 팀, 전화번호, 이메일, 직책과 내가 참여 했던 프로젝트 목록을 확인할 수 있어야 한다.	기능
REQ-58	프로필	내 정보 관리	모든 사용자	내 정보 수정	사용자의 이름, 팀, 전화번호, 이메일, 직책을 수정할 수 있어야 한다.	기능
REQ-59	프로필	내 프로젝트 목록	모든 사용자	프로젝트 목록 조회	사용자가 참여 했던 프로젝트 목록을 조회할 수 있어야 한다.	기능
REQ-60	프로필	내 정보 관리	모든 사용자	내 정보 접근	권한이 없는 사용자가 정보를 수정, 삭제할 경우 접근 차단 해야 한다.	비기능
REQ-61	프로필	내 프로젝트 목록	모든 사용자	프로젝트 상세 보기	프로젝트 목록중에 참여한 프로젝트의 릴리즈 노트를 확인할 수 있어야한다.	기능

REQ-62	릴리즈 노트	릴리즈 노트 목록 조회	프로젝트 멤버	릴리즈 노트 조회	버전, 상태, 진행률, 생성일자, 릴리즈 일자, 해당 릴리즈노트 설명이 표 형식으로 노출 되어야한다.	기능
REQ-63	릴리즈 노트	릴리즈 노트 목록 조회	프로젝트 멤버	릴리즈 노트 상태, 진행률 변경	릴리즈 노트의 상태(릴리즈됨, 릴리즈 안됨)가 표시되어야 하며, 작성자와 작성 일자, 버전이 노출 되어야 한다.	기능
REQ-64	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 조회	릴리즈 노트에 대한 설명, 관련 작업 첨부, 관련 이슈, 릴리즈 일자, 담당자, 진행률을 조회 할 수 있어야 한다.	기능
REQ-65	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 조회	페이지 로드 시에 해당 릴리즈 노트의 정보를 HTTP GET 으로 가져와서 올바르게 표시해야 한다.	기능
REQ-66	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 조회	페이지 로드 시에 해당 릴리즈 노트에 등록되어 있는 이슈 목록을 HTTP GET 으로 가져와서 올바르게 표시해야 한다.	기능
REQ-67	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 조회	표시해야 하는 모든 정보를 HTTP GET 으로 가져와서 렌더링 할 정보와 함께 로딩이 완료되는 데에 3 초를 초과하지 않아야 한다.	비기능
REQ-68	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 조회	<ul style="list-style-type: none"> - 진행률을 나타내는 Input 칸에는 0 부터 100 까지의 수만 입력될 수 있으며, 0 이하의 수가 입력되면 0 이, 100 이상이 입력되면 100 이, 숫자가 아닌 문자가 입력되면 무시해야 한다. - 진행률을 나타내는 Input 칸에 입력되는 수는 입력과 동시에 칸 하단의 진행 바에 실시간으로 반영되어야 하며, 진행률에 따라서 색깔이 변경되어야 한다. 	기능
REQ-69	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 이슈 조회	- 릴리즈 노트에 관련되어 있는 이슈 목록이 표시되며, 담당자 필터와 상태(Backlog, In Progress, Done) 필터를 적용시켜 간편하게 조회할 수 있다.	기능

					- 필터 초기화 버튼을 통해 필터를 초기화하고 릴리즈 노트에 관련되어 있는 모든 이슈를 확인할 수 있다.	
REQ-70	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 이슈 추가/삭제	<ul style="list-style-type: none"> - 이슈 표에서 추가하기 버튼을 눌러 해당 릴리즈 노트에 이슈를 추가할 수 있으며, 추가 후에는 추가하기 모달 창에서는 해당 이슈가 보이지 않고 관련된 이슈 창에서는 추가되어야 한다. - 이슈 표에서 이슈 클릭 후 제거하기 버튼을 누르면 해당 릴리즈 노트에서 이슈가 제거되며, 관련된 이슈 창에서 해당 이슈가 나타나지 않아야 한다. 이슈 추가하기 모달 창에서 해당 이슈를 다시 확인하고 추가할 수 있어야 한다. 	기능
REQ-71	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 이슈 추가/삭제	<ul style="list-style-type: none"> - 릴리즈 노트 생성/수정 페이지에서 이슈가 추가되면 그 즉시 이슈 표에서 확인할 수 있어야 하고, 이슈 추가하기 모달 창에서는 사라져야 한다. - 릴리즈 노트 생성/수정 페이지에서 이슈가 삭제되면 그 즉시 이슈 표에서 사라져야 하고, 이슈 추가하기 모달 창에서 나타나야 한다. 	비기능
REQ-72	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 수정	릴리즈 노트 수정 후 '수정' 버튼을 클릭하면 릴리즈 노트의 정보들과 관련된 이슈 목록을 HTTP PUT 요청을 전달한 후 릴리즈 노트 목록 페이지로 이동한다.	기능
REQ-73	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 수정	'수정' 버튼을 눌러 HTTP PUT 요청을 제출하고 릴리즈 노트 목록 페이지로 이동 완료하는 데에 3 초를 초과하지 않아야 한다.	비기능
REQ-74	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 생성	내용 입력 후 '저장' 버튼을 누르면 HTTP POST 요청을 통해 정보를 업로드 하고 릴리즈 노트 목록 페이지로 이동한다.	기능
REQ-75	릴리즈 노트	릴리즈 노트 내용 조회	프로젝트 멤버	릴리즈 노트 생성	'저장' 버튼을 눌러 HTTP PUT 요청을 제출하고 릴리즈 노트 목록 페이지로 이동 완료하는 데에 3 초를 초과하지 않아야 한다.	비기능

REQ-76	릴리즈 노트	릴리즈 노트 내에서 관련 이슈 조회	모든 사용자	릴리즈 노트 내에서 관련 이슈 조회	릴리즈 노트 조회시 관련 이슈를 클릭하면 , 이슈 관련 세부정보를 조회 할 수 있어야 한다.	기능
REQ-77	릴리즈 노트	릴리즈 노트 생성(추가)	PM	릴리즈 노트 생성	릴리즈 노트에 대한 설명, 관련 작업 첨부, 관련 이슈를 생성할 수 있어야 한다.	기능
REQ-78	릴리즈 노트	릴리즈 노트	PM	릴리즈 노트 수정	릴리즈 노트에 대한 설명, 관련 작업 첨부, 관련 이슈를 수정할 수 있어야 한다.	기능
REQ-79	릴리즈 노트	릴리즈 노트	PM	릴리즈 노트 삭제	릴리즈 노트 목록에서 해당 릴리즈 노트를 삭제할 수 있어야 한다. 삭제 요청 시 정말 삭제할 것인지 재확인할 수 있도록 한다.	기능
REQ-80	프로젝트 관리	프로젝트 관리	PM	프로젝트 정보 조회	프로젝트 이름, 설명, 그룹, 상태, 생성일을 조회할 수 있다.	기능
REQ-81	프로젝트 관리	프로젝트 관리	PM	프로젝트 멤버 관리	프로젝트에 참여할 멤버를 추가, 삭제할 수 있다.	기능
REQ-82	프로젝트 관리	프로젝트 관리	PM	프로젝트 멤버 추가	프로젝트 멤버를 추가할 때 그룹 안에 있는 멤버를 검색할 수 있어야 한다.	기능
REQ-83	프로젝트 관리	프로젝트 관리	PM	PM 권한 부여	프로젝트 멤버 중 PM 권한을 양도할 수 있다.	기능
REQ-84	프로젝트 관리	프로젝트 관리	PM	프로젝트 수정	프로젝트 이름, 설명, 그룹, 상태, 생성일을 수정할 수 있다.	기능
REQ-85	프로젝트 관리	프로젝트 관리	PM	프로젝트 멤버 조회	프로젝트에 참여하고 있는 멤버의 리스트를 조회할 수 있다.	기능
REQ-86	로그아웃	로그아웃	그룹 멤버	로그아웃	- 로그아웃 시 Access_Token 을 삭제 한다. - 로그인 화면에서 재 로그인 시 다시 서비스를 이용할 수 있다.	기능
REQ-87	웹서버	NginX		정적 파일 서빙 기능 제공	Nginx 를 사용하여 정적 파일 서빙을 지원해야 한다.	기능
REQ-88	웹서버	NginX		웹 애플리케이션 서버와 연결	웹 애플리케이션 서버(Tomcat)와의 연결을 설정해야 한다.	기능

REQ-89	웹서버	NginX		HTTPS 지원	SSL/TLS 설정을 포함하여 HTTPS 를 지원해야 한다.	기능
REQ-90	웹서버	NginX		도메인 및 서브도메인 설정	다중 도메인 호스팅이 가능하도록 도메인 및 서브도메인 설정을 지원한다.	기능
REQ-91	웹 애플리케이션 서버	Tomcat		Java 웹 애플리케이션 호스팅	Java 웹 애플리케이션을 호스팅하기 위해 Tomcat 을 사용한다.	기능
REQ-92	웹 애플리케이션 서버	Tomcat		데이터베이스 연동	JDBC 드라이버 설정으로 데이터베이스와 연동한다.	기능
REQ-93	웹 애플리케이션 서버	Tomcat		고급 기능 지원	세션 관리, 보안 설정, 클러스터링 등 고급 기능이 필요하다.	기능
REQ-94	웹서버, 웹 애플리케이션 서버	Nginx, Tomcat		프록시 설정	Nginx 를 통해 웹 애플리케이션 서버로 요청을 전달하는 프록시를 설정한다.	기능
REQ-95	데이터베이스	ElasticSearch, MySQL		데이터베이스 연결	-MySQL 을 활용하여 정형화된 데이터 저장 및 관리한다. -엘라스틱서치를 활용하여 복잡한 검색 기능을 구현한다.	기능
REQ-96	제품	운영체제		리눅스 사용	서버 운영체제는 리눅스 기반 시스템을 사용해야 한다.	비기능
REQ-97	조직	구현 방법론		React	React 를 기반으로하여 웹페이지를 구현한다.	기능
REQ-98	제품	이식성		브라우저	브라우저는 크롬을 기준으로 한다.	비기능
REQ-99	조직	프로그래밍 언어		JavaScript Java 사용	프로그래밍 언어는 FrontEnd 에서는 JavaScript, Backend 에서는 Java 를 활용한다.	기능

5. 기술 스택 및 버전

1) 개발 환경

Kakao I Cloud

2) 개발 도구

intelliJ (Spring boot)

Visual Studio Code (React)

1) 개발 언어

Front-end : Javascript, CSS

Back-end : Java (JDK Version : open-jdk-17)

2) 개발 프레임 워크

Web server : Nginx

Web Application Server : Spring boot

Database : Mysql , Redis

infra : Docker

3) 형상 관리 시스템

CI/CD : Jenkins

협업 Tool

GitHub

4) 주요 Library

Front-end: react, mui, bootstrap

Back-end: JsonWebToken

6. 코드 컨벤션

1) Directory Format

- 도메인 별로 구분
- 각 도메인 별로 controller, service, repository, DTO를 분리해서 사용

2) Naming

- 패키지 이름은 **소문자** 사용

```
package kakao99.backend.example;
```

- 클래스/인터페이스는 **명사, UpperCamelCase** 사용

```
public class ReleaseNote{  
    ...  
}
```

- 상수(Enum)은 **CONSTANT_CASE** 사용

```
private static final NOT_MATCH_CODE
```

- dto 객체는 이름 뒤에 대문자 **DTO** 사용

```
public class ReleaseNoteDTO {  
    ...  
}
```

- dto 객체가 Request Body에 사용되면 **RequestDTO**, Resposne Body에 담아서 응답하면 **ResponseDTO**로 사용

```
@PostMapping("/")  
public MemberResponseDTO getMemberInfo(@RequestBody MemberRequestDTO memberRequestDTO) {  
    ... (생략)  
  
    return memberResponseDTO;  
}
```

- 메서드 이름에 **동사**, **lowerCamelCase** 사용

```
public Member getMemberInfo(){  
    ...  
}
```

- JPA, queryDSL 메소드 이름을 **find이름By이름..**로 사용한다.

```
@Repository  
public class MemberRepository{  
  
    private final EntityManager em;  
    private final JPAQueryFactory query;  
    ... (생략)  
  
    public Optional<Member> findMemberByMemberId(Long memberId) {  
        ... (생략)  
    }  
}
```

- JPA, queryDSL을 제외한 다른 클래스의 메소드 이름에서 find로 시작하는 이름을 쓰지 않는다.

- 파라미터 이름을 사용할 때 **두글자 이상** 사용해야 하며 파라미터의 의미가 **명확**해야 한다.

```
//id가 명확하지 않음  
public Member findById(Long id);  
  
//id가 명확함  
public Member findById(Long memberId);
```

- Test 코드는 클래스명 뒤에 Test를 붙여 사용한다.

3) Naming

- Entity에 **Setter**를 사용하지 않는다.

```
@Getter  
@Entity  
public class Member {  
    ... (생략)  
}
```

- 클래스 생성자는 **Builder** pattern을 사용한다.

```
@Getter  
@Entity  
@Builder  
public class Member {  
    ... (생략)  
}
```

- DI 주입은 생성자 주입으로 사용하되, `@RequiredArgsConstructor` 를 사용한다.

```
@Service
@RequiredArgsConstructor
public class MemberService {
    private final MemberRepository memberRepository;
    ... (생략)
}
```

- Controller에 Service를 의존성 주입하고 Service에 Repository를 주입한다.

```
@Service
@RequiredArgsConstructor
public class MemberService {
    private final MemberRepository memberRepository;
    ... (생략)
}

@RestController
@RequiredArgsConstructor
public class MemberController {
    private final MemberService memberService;
    ... (생략)
}
```

- 예외 처리는 Service에서 진행하고 Service에는 항상 DTO만 반환하며 Controller에서 `ResponseEntity`를 생성하여 사용한다.

```
@Service
public class MemberService {
    ... (생략)

    public MemberInfoDTO getMemberInfo(Long memberId) {

        Optional<Member> member = memberRepository.findMemberByMemberId(memberId);
        if ( member.isEmpty() ) {
            throw new Exception...
        }

        ... (생략)

        return memberInfoDTO;
    }
}
```

- 컬렉션 인터페이스는 변수명 뒤에 컬렉션 종류를 붙여서 선언한다.

```
//잘못된 예
List<Member> members;

//잘된 예
List<Member> memberList;
```

- 클래스 선언, 메서드 선언, 조건/반복문 등의 코드 블록을 감싸는 중괄호는 **K&R 스타일**을 사용한다.

```
//잘못된 예
if (member == null)
{
    return null;
}

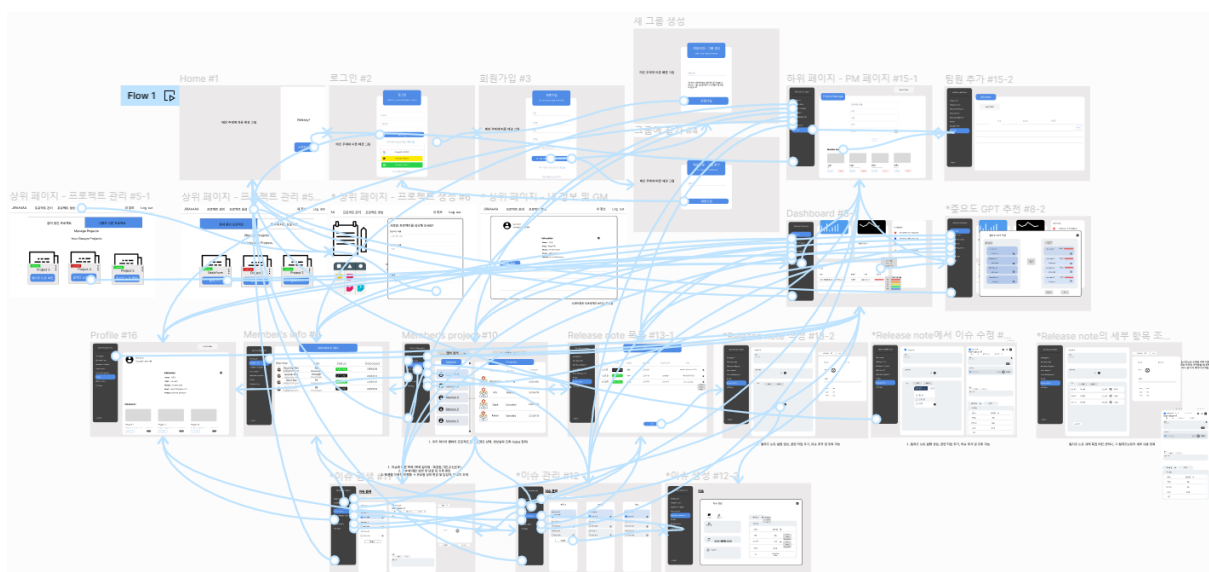
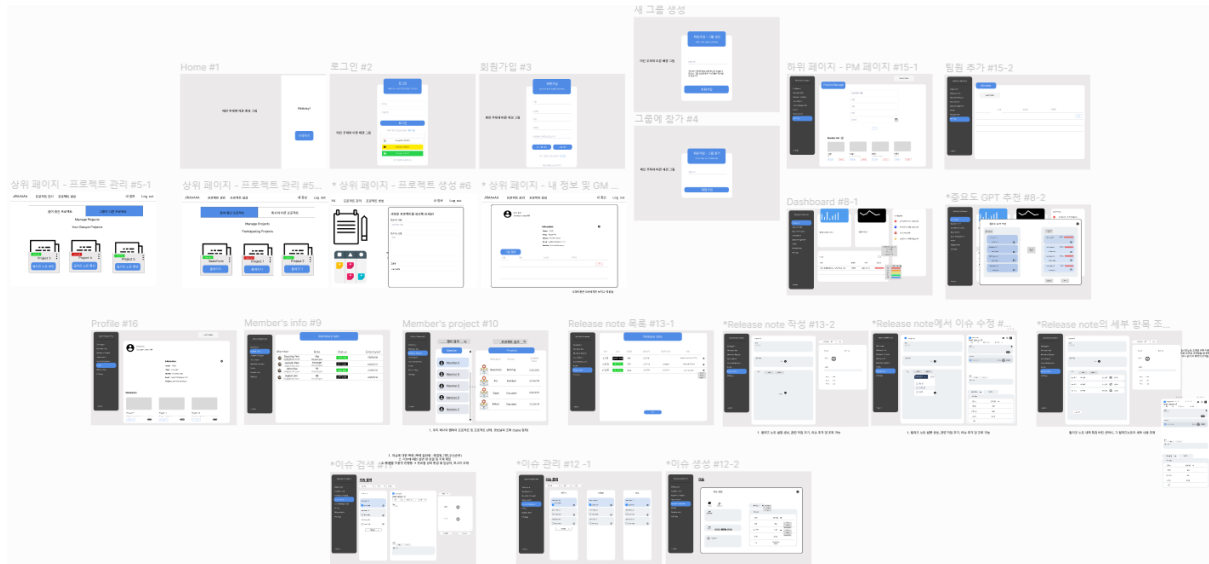
//잘된 예
if (member == null) {
    return null;
}
```

- 중괄호 사용시 한줄 띄어서 작성한다.
- 닫는 중괄호와 같은 줄에 else, catch, finally, while 선언한다.
- 조건/반복문에 **중괄호** 반드시 사용한다.
- 조건/반복문 뒤에 오는 괄호()는 공백 후 사용한다.
- 클래스를 import할 때 와일드카드(*)를 사용하지 않는다.
- 로그는 System.out.println() 대신 **Slf4j**를 사용한다.
- 필드 값 주석은 변수 옆에 쓰고, 상세 설명 같은 주석은 바로 위에 작성한다.
- return 위에 한 줄을 띄어서 작성한다.
- Test 코드에는 **when, given, then** 에 맞춰 작성한다.

4) Gradle

주석으로 dependency에 대한 설명을 작성한다.

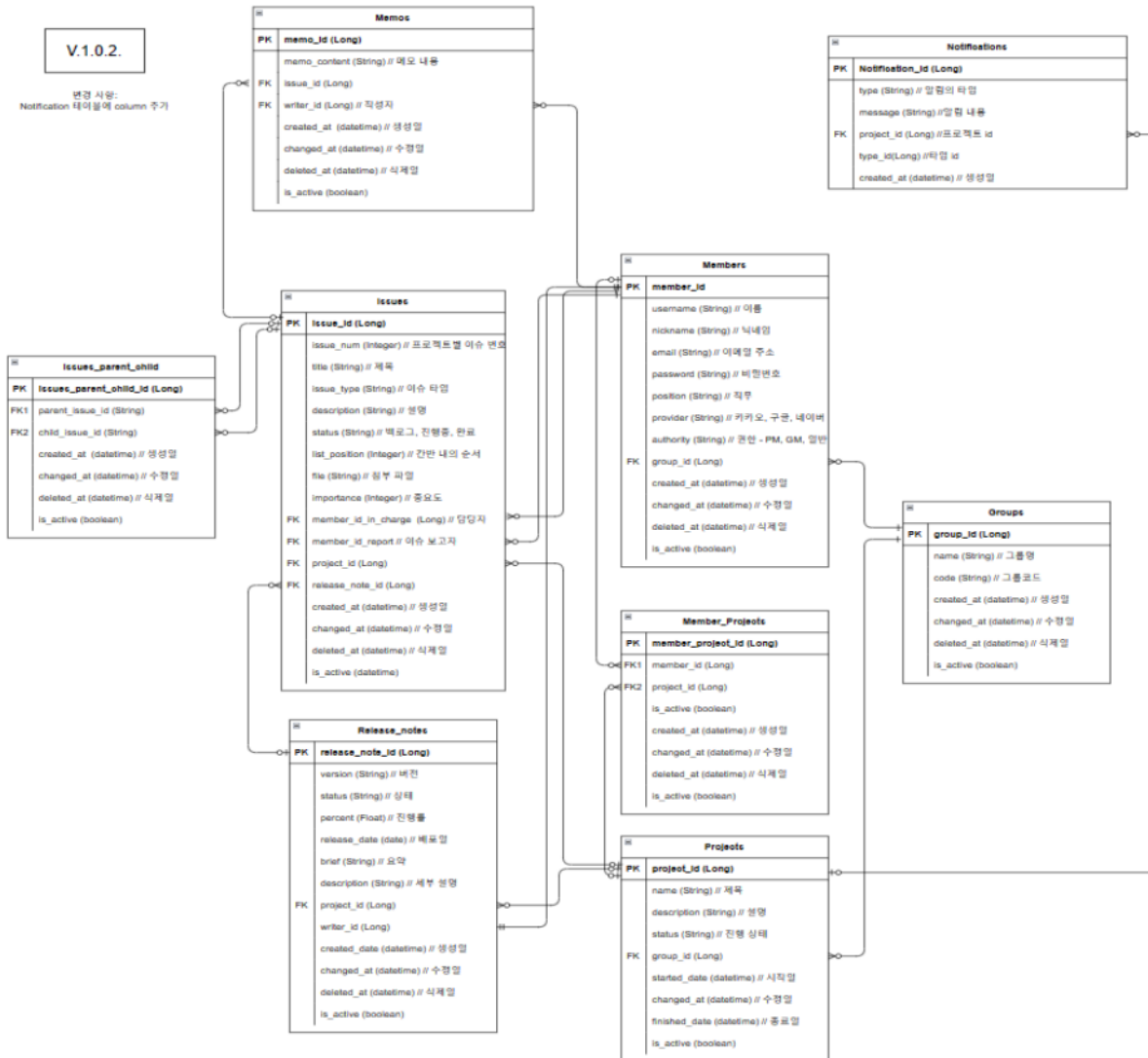
7. 와이어프레임 및 스토리보드



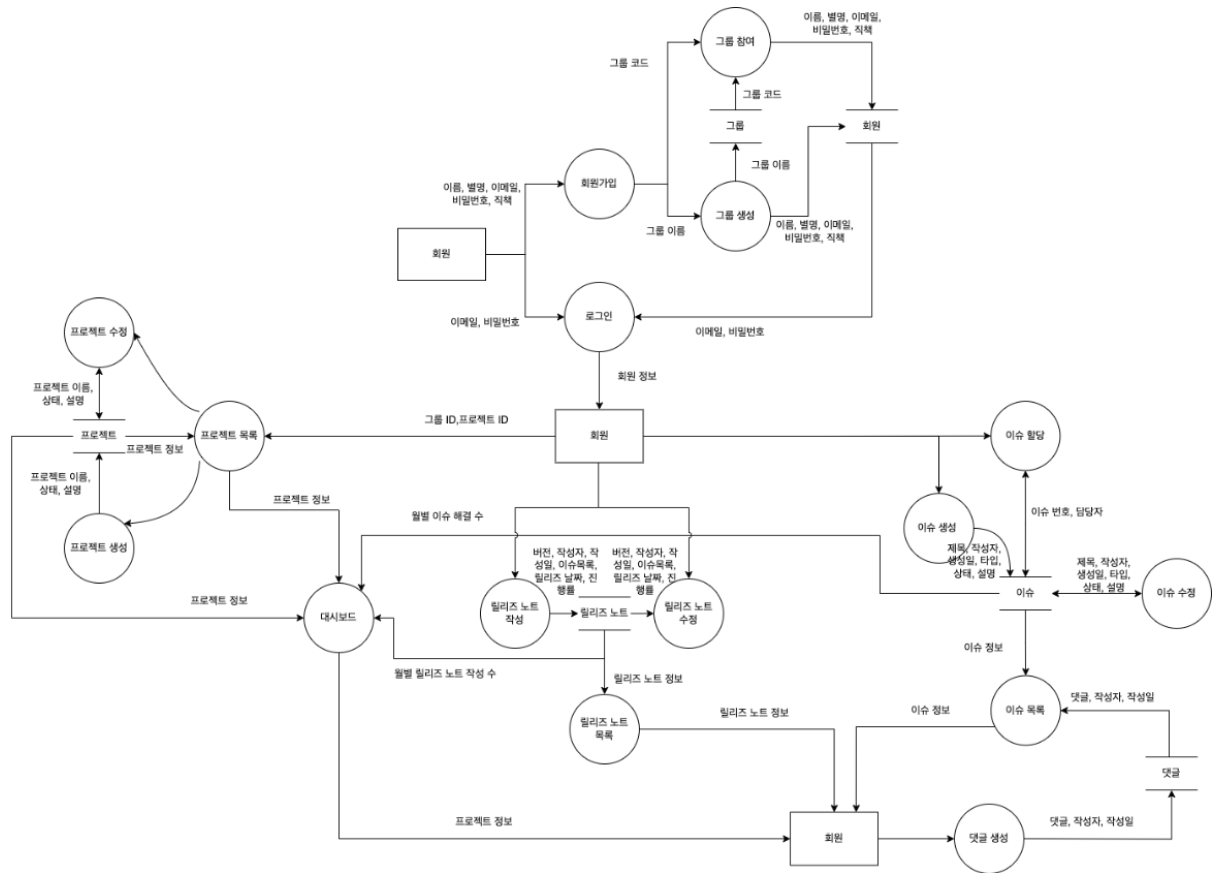
8. 서비스 실행 화면

추가 예정

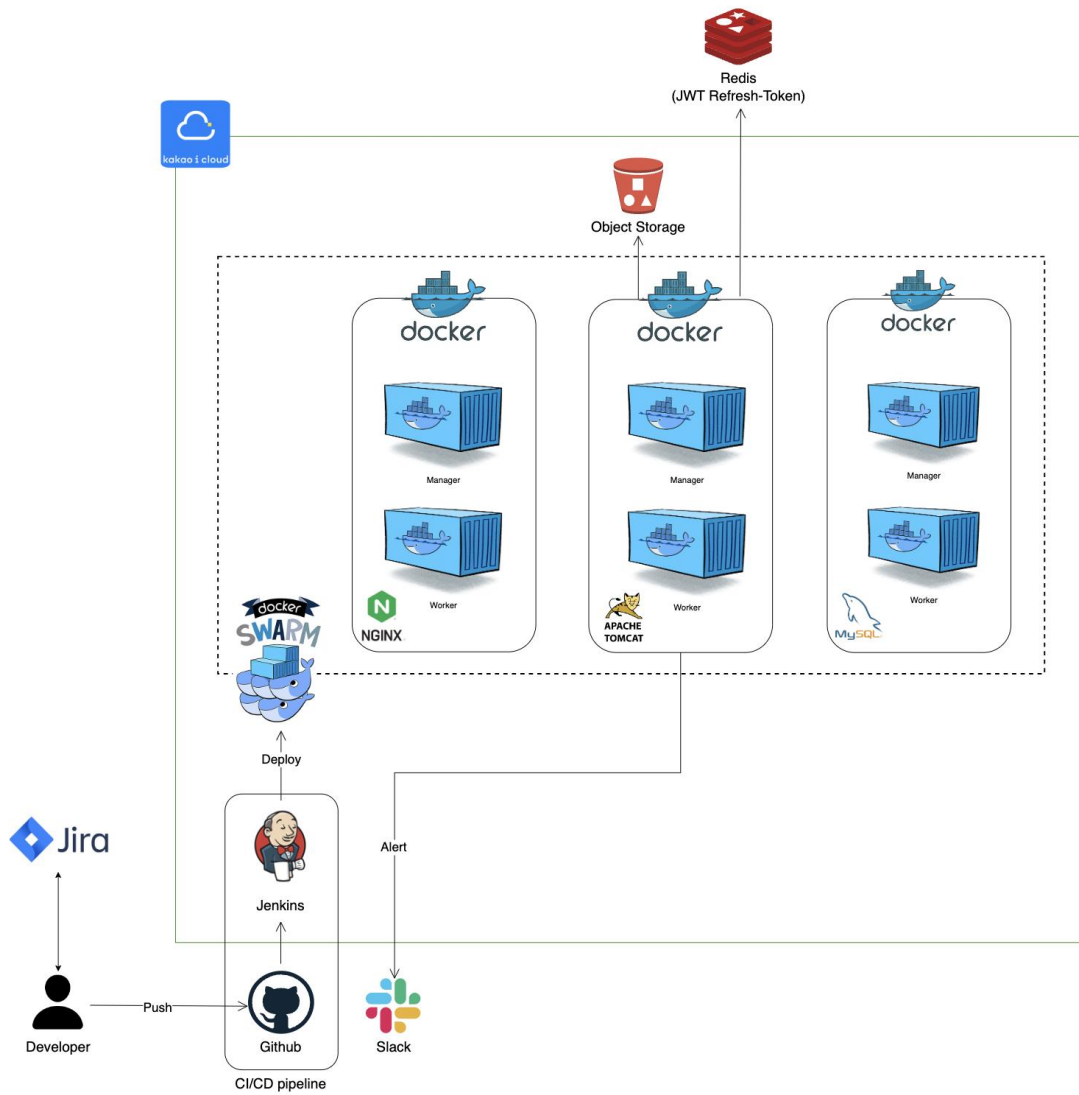
9. 개체-관계 다이어그램(ERD)



10. 데이터 흐름도(DFD)



11. 아키텍처 구조도



12. API 및 인터페이스 명세서

1) 회원

회원가입 - 그룹 생성

설명	회원가입 시 그룹을 생성한다.				
URL	/api/member/signup/group				
Method	POST		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Body	name	Y	N	String	사용자 이름
	nickname	Y	N	String	사용자 별명
	provider	N	N	String	소셜 로그인 제공자
	email	Y	N	String	이메일
	password	Y	N	String	비밀번호
	position	Y	N	String	직책
	groupName	Y	N	String	생성할 그룹 이름
{ "name": "홍길동", "nickname": "hong", "provider": "google", "email": "kakao99@google.com", "password": "12345678", "position": "FE", "groupName": "kakao99" }					
Response Message					
HTTP	항목	반복여부		type	설명
Body	statusCode	N		Integer	결과코드 200 이외에는 전부 에러
	message	N		String	정상결과/에러발생시의 메시지
{ "statusCode": 200, "message": "회원 가입이 완료 되었습니다." }					

회원가입 – 그룹 참가

설명	회원가입 시 그룹을 생성한다.				
URL	/api/member/signup/group/join				
Method	POST		Content-Type		application/json;charset=UTF-8
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Body	name	Y	N	String	사용자 이름
	nickname	Y	N	String	사용자 별명
	provider	N	N	String	소셜 로그인 제공자
	email	Y	N	String	이메일
	password	Y	N	String	비밀번호
	position	Y	N	String	직책
	groupName	Y	N	String	참가할 그룹 코드
{ "name": "홍길동", "nickname": "hong", "provider": "google", "email": "kakao99@google.com", "password": "12345678", "position": "FE", "groupName": "23ed6106-d9c3-4260-9413-b668e9d0761a" }					
Response Message					
HTTP	항목	반복여부		type	설명
Body	statusCode	N		Integer	결과코드 200 이외에는 전부 에러
	message	N		String	정상결과/에러발생시의 메시지
{ "statusCode": 200, "message": "회원 가입이 완료되었습니다." }					

로그인

설명	이메일과 비밀번호로 로그인한다.				
URL	/api/member/login				
Method	POST		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수 여부	반복 여부	type	설명
Body	email	Y	N	String	이메일
	password	Y	N	String	비밀번호
{ "email": "kakao99@google.com", "password": "12345678" }					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
	data		N	String	
		accessToken	N	String	JWT 토큰
{ "statusCode": 200, "message": "회원 가입이 완료 되었습니다.", "data": [{ "accessToken" : "eyJhbG5kdse...gyanzrFxnKvSvDrg" }] }					

회원 정보 조회

설명	회원 정보와 회원이 참여한 프로젝트 목록을 조회한다.				
URL	/api/member				
Method	GET	Content-Type	application/json;charset=UTF-8		
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	Authorization	Y	N	String	로그인시 부여 받은 Access Token
Authorization : Bearer eyJhbG5SWdse...gyanzrFxnKvSvDrg					
Response Message					
HTTP	항목	반복여부		type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
	data		N	Object	
	name		N	String	로그인한 사용자 이름
	nickname		N	String	로그인한 사용자 별명
	email		N	String	로그인한 사용자 이메일
	groupName		N	String	로그인한 사용자가 속한 그룹 이름
	position		N	String	로그인한 사용자의 직책
	introduce		N	String	로그인한 사용자의 소개
	projectList		N	Object	로그인한 사용자가 참여한 프로젝트 목록
		id	N	String	프로젝트 id
		name	N	String	프로젝트 이름
		description	N	String	프로젝트 소개
		status	N	String	프로젝트 상태
		createdAt	N	String	프로젝트 시작일
		updatedAt	N	String	프로젝트 수정일
		deletedAt	N	String	프로젝트 삭제일
		isActive	N	String	삭제 여부


```
{
  "statusCode": 200,
  "message": "회원 정보 조회 완료 되었습니다.",
  "data": [
    {
      "name" : "홍길동",
      "nickname" : "hong",
      "email" : hong@google.com,
      "groupName" : "kakao99",
      "position" : "FE",
      "introduce" : "안녕하세요 자기소개입니다.",
      projectList : [
        {
          "id" : "1",
          "name" : "BrainForm",
          "description" : "프로젝트 소개",
          "status" : "stopped",
          "createdAt" : "2023-06-23",
          "updatedAt" : "2023-06-29",
          "deletedAt" : null,
          "isActive" : "true"
        }
      ]
    }
  ]
}
```

회원 정보 수정

설명	회원 정보를 수정한다.				
URL	/api/member				
Method	Patch		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	Authorization	Y	N	String	로그인시 부여 받은 Access Token
Body	name	Y	N	String	수정할 이름
	nickname	N	N	String	수정할 별명
	position	Y	N	String	수정할 직책
	introduce	Y	N	String	수정할 소개
Authorization : Bearer eyJhbGVSdse...gyanzrFxnKvSvDrg					
{ "name": "이순신", "nickname": "Lee", "position": "BE", "introduce" : "안녕하세요 수정한 자기소개입니다." }					
Response Message					
HTTP	항목	반복여부		type	설명
Body	statusCode	N		Integer	결과코드 200 이외에는 전부 에러
	message	N		String	정상결과/에러발생시의 메시지
{ "statusCode": 200, "message": "멤버 정보 수정이 완료 되었습니다." }					

프로젝트 내의 회원 조회

설명	한 프로젝트에 참여하고 있는 멤버들을 조회한다.				
URL	/api/project/{projectId}/members				
Method	GET			Content-Type	application/json;charset=UTF-8
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	Authorization	Y	N	String	로그인시 부여 받은 Access Token
Param	projectId	Y	N	String	프로젝트 Id
Authorization : Bearer eyJhbGVSdse...gyanzrFxnKvSvDrg					
Response Message					
HTTP	항목	반복여부		type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
	data				
	memberInfoDTOList				
		name	N	String	회원 이름
		nickname	N	String	회원 별명
		email	N	String	회원 이메일
		position	N	String	회원 직책
{ "statusCode": 200, "message": "회원 정보 조회 완료 되었습니다.", "data": [{ memberInfoDTOList : ["name" : "홍길동", "nickname" : "hong", "email" : "hong@google.com", "position" : "BE"] }] }					

2) 이슈

이슈 생성

설명	프로젝트에 해당하는 이슈를 생성한다.				
URL	/api/project/{projectId}/issues				
Method	POST	Content-Type	application/json;charset=UTF-8		
Request Message					
HTTP	항목	필수여부	반복 여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
param	projectId	Y	N	Long	이슈를 포함하는 프로젝트 id
body	id			Long	이슈 id
	issueNum			Int	이슈 넘버
	title			String	이슈 제목
	issueType			String	이슈 타입(task,bug,story)
	description			String	이슈 세부사항 설명
	status			String	이슈 상태 (backlog,inprogress,done)
	listPosition			Float	칸반 노트 순서
	importance			Int	중요도(%)
	file			String	첨부 파일
	createdAt			Date	생성 일자
	memberIdInCharge			Object	담당자 정보
		name		String	담당자 이름
		nickname		String	담당자 닉네임
	memberReport			Object	보고자 정보
		Name		String	보고자 이름
		nickname		String	보고자 닉네임
		position		String	보고자 직무(FE,BE)
Authorization : Bearer eyJqhdXic1J9.eyJ3Z.....HSajskzqodu					
{ "id": 1, "issueNum": 1, "title": "제목 수정", "issueType": "bug", "description": "이슈 세부사항", "status": "backlog", "importance": null,					

```

    "file": "test_file_path",
    "createdAt": "2023-07-12T06:28:09.567+00:00",
    "memberIdInCharge": {
      "name": "박재석",
      "nickname": "박재석",
      "email": "qkrwotjr0@gmail.com",
      "position": "FE"
    },
    "memberReport": null
  }

```

Response Message

HTTP	항목			반복 여부	type	설명
	statusCode			N	Int	결과코드 200 이외에는 전부 에러
	message			N	Long	메시지
	data			N	Object	전송되는 데이터

```

{
  "statusCode": 200,
  "message": "1번 이슈 생성 성공",
  "data": [ ]
}

```

이슈 조회

	프로젝트에 해당하는 모든 이슈를 조회한다. -param에 따라서 필터링을 통해 이슈를 조회할 수 있다. -excludeId로 하위 이슈로 연결된 id를 제외하고 조회할 수 있다.				
URL	/api/project/{projectId}/issues				
Method	GET			Content-Type	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Param	projectId	Y	N	Long	이슈를 포함하는 프로젝트 id
	status	N	N	String	이슈 상태로 필터링할 값 (backlog, inprogress,done)
	type	N	N	String	이슈 타입으로 필터링할 값(bug, task, story)
	username	N	N	String	해당 이슈 담당자로 필터링할 값
	excludeId	N	N	Long	선택한 이슈 id(하위 이슈를 조회 시 자신은 제외하여야 하기 때문)
Authorization : Bearer eyJqhdXic1J9.eyJ3z.....HSajskzqodu					
Response Message					
HTTP	항목		반복여부	type	설명
	statusCode			N Int	결과코드 200 이외에는 전부 에러
	message			N Long	메시지
	data			N Object	전송되는 데이터
		id		Long	이슈 id
		issueNum		Int	이슈 넘버
		title		String	이슈 제목
		issueType		String	이슈 타입(task,bug,story)
		description		String	이슈 세부사항 설명
		status		String	이슈 상태 (backlog,inprogress,done)
		listPosition		Float	칸반 노트 순서
		importance		Int	중요도(%)
		file		String	첨부 파일
		createdAt		Date	생성 일자
		releasenote		String	릴리즈노트 버전 이름
		memberIdInCharge		Object	담당자 정보

			name		String	담당자 이름
			nickname		String	담당자 닉네임
			email		String	담당자 이메일
			position		String	담당자 직무(FE,BE)
		memberReport			Object	보고자 정보
			Name		String	보고자 이름
			nickname		String	보고자 닉네임
			email		String	보고자 이메일
			position		String	보고자 직무(FE,BE)

```
{
  "statusCode": 200,
  "message": "1번 프로젝트의 모든 이슈 상태,타입,담당자별 조회 성공",
  "data": [
    {
      "id": 1,
      "issueNum": 1,
      "title": "제목 수정",
      "issueType": "bug",
      "description": "이슈 세부사항",
      "status": "backlog",
      "listPosition": null,
      "importance": null,
      "file": "test_file_path",
      "createdAt": "2023-07-12T06:28:09.567+00:00",
      "releasenote": "v1.0.0",
      "memberIdInCharge": {
        "name": "박재석",
        "nickname": "박재석",
        "email": "qkrwotjr0@gmail.com",
        "position": "FE"
      },
      "memberReport": null
    },
  ],
}
```

이슈 수정

설명	프로젝트에 해당하는 이슈를 수정한다.				
URL	/api/project/{projectId}/issues/{issueId}				
Method	PUT		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Param	projectId	Y	N	Long	이슈를 포함하는 프로젝트 id
Body	id			Long	이슈 id
	issueNum			Int	이슈 넘버
	title			String	이슈 제목
	issueType			String	이슈 타입(task,bug,story)
	description			String	이슈 세부사항 설명
	status			String	이슈 상태 (backlog,inprogress,done)
	listPosition			Float	칸반 노트 순서
	importance			Int	중요도(%)
	file			String	첨부 파일
	createdAt			Date	생성 일자
	releasenote			String	릴리즈노트 버전 이름
	memberIdInCharge			Object	담당자 정보
		name		String	담당자 이름
		nickname		String	담당자 닉네임
		email		String	담당자 이메일
		position		String	담당자 직무(FE,BE)
	memberReport			Object	보고자 정보
		Name		String	보고자 이름
		nickname		String	보고자 닉네임
		email		String	보고자 이메일
		position		String	보고자 직무(FE,BE)

Authorization : Bearer eyJqhdXic1J9.eyJ3z.....HSajskzqodu

{

 "id": 1,
 "issueNum": 1,
 "title": "제목 수정",
 "issueType": "bug",


```

    "description": "이슈 세부사항",
    "status": "backlog",
    "importance": null,
    "file": "test_file_path",
    "createdAt": "2023-07-12T06:28:09.567+00:00",
    "memberIdInCharge": {
      "name": "박재석",
      "nickname": "박재석",
      "email": "qkrwotjr0@gmail.com",
      "position": "FE"
    },
    "memberReport": null
  }

```

Response Message

HTTP	항목			반복 여부	type	설명
	statusCode			N	Int	결과코드 200 이외에는 전부 에러
	message			N	Long	메시지
	data			N	Object	전송되는 데이터

```

{
  "statusCode": 200,
  "message": "1번 이슈 생성 성공",
  "data": [ ]
}

```

이슈 삭제

설명	해당 이슈를 삭제한다.					
URL	/api/project/{projectId}/issues					
Method	DELETE			Content-Type		
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 부여 받은 Access Token
Param	projectId		Y	N	Long	이슈를 포함하는 프로젝트 id
Authorization : Bearer eyJqhdlxic1J9.eyJ3Z.....HSajskzqodu						
{ "id" = 1, }						
Response Message						
HTTP	항목			반복여부	type	설명
	statusCode			N	Int	결과코드 200 이외에는 전부 에러
	message			N	Long	메시지
	data			N	Object	전송되는 데이터
{ "statusCode": 200, "message": "1번 이슈 삭제 성공", "data": [] }						

하위 이슈 생성

설명	선택한 이슈에 대한 하위 이슈를 생성한다.					
URL	/api/project/{projectId}/issues/{issuelId}/childissue					
Method	POST		Content-Type	application/json;charset=UTF-8		
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 부여 받은 Access Token
Param	projectId		Y	N	Long	이슈를 포함하는 프로젝트 id
	issuelId		Y	N	Long	선택한 이슈의 id
Body	parentIssuelId		Y	N	Long	상위 이슈로 부여된 id
	childIssuelId		Y	N	Long	하위 이슈로 부여된 id
	createdAt		Y	N	String	생성일자
	updatedAt		N	N	String	수정일자
	deletedAt		N	N	String	삭제일자
Authorization : Bearer eyJqhdXic1J9.eyJ3Z.....HSajskzqodu						
{ "parentIssuelId " = 1, "childIssuelId" = 2, }						
Response Message						
HTTP	항목			반복여부	type	설명
	statusCode			N	Int	결과코드 200 이외에는 전부 에러
	message			N	Long	메시지
	data			N	Object	전송되는 데이터
{ "statusCode": 200, "message": "1번 하위 이슈 생성 성공", "data": [] }						

하위 이슈 수정

설명	선택한 이슈에 대한 하위 이슈를 수정한다.					
URL	/api/project/{projectId}/issues/{issueId}/childissue					
Method	PUT		Content-Type	application/json;charset=UTF-8		
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 부여 받은 Access Token
Param	projectId		Y	N	Long	이슈를 포함하는 프로젝트 id
	issueId		Y	N	Long	선택한 이슈의 id
Body	parentIssueId		Y	N	Long	상위 이슈로 부여된 id
	childIssueId		Y	N	Long	하위 이슈로 부여된 id
	createdAt		Y	N	String	생성일자
	updatedAt		N	N	String	수정일자
	deletedAt		N	N	String	삭제일자
Authorization : Bearer eyJqhdXic1J9.eyJ3Z.....HSajskzqodu						
{ "parentIssueId " = 1, "childIssueId" = 2, }						
Response Message						
HTTP	항목		반복여부	type	설명	
	statusCode		N	Int	결과코드 200 이외에는 전부 에러	
	message		N	Long	메시지	
	data		N	Object	전송되는 데이터	
{ "statusCode": 200, "message": "1번 하위 이슈 수정 성공", "data": [] }						

하위 이슈 삭제

설명	선택한 이슈에 대한 하위 이슈를 삭제한다.					
URL	/api/project/{projectId}/issues/{issueId}/childissue					
Method	DELETE		Content-Type	application/json;charset=UTF-8		
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 부여 받은 Access Token
Param	projectId		Y	N	Long	이슈를 포함하는 프로젝트 id
	issueId		Y	N	Long	선택한 이슈의 id
Body	parentIssueId		Y	N	Long	상위 이슈로 부여된 id
	childIssueId		Y	N	Long	하위 이슈로 부여된 id
	createdAt		Y	N	String	생성일자
	updatedAt		N	N	String	수정일자
	deletedAt		N	N	String	삭제일자
Authorization : Bearer eyJqhdXic1J9.eyJ3Z.....HSajskzqodu						
{ "parent_child_IssueId " = 1, }						
Response Message						
HTTP	항목		반복여부	type	설명	
	statusCode			N	Int	결과코드 200 이외에는 전부 에러
	message			N	Long	메시지
	data			N	Object	전송되는 데이터
{ "statusCode": 200, "message": "1번 하위 이슈 삭제 성공", "data": [] }						

3) 댓글

댓글 생성

설명	해당 이슈에 대한 댓글을 생성한다.				
URL	/api/project/{projectId}/issues/{issueId}/memo				
Method	POST		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Param	projectId	Y	N	Int	이슈를 포함하는 프로젝트 id
	issueId	Y	N	Int	댓글을 포함하는 이슈 id
Body	issueId	Y	N	Int	댓글을 포함하는 이슈 id
	content	N	N	String	댓글 내용
	createAt	Y	N	Date	댓글 생성일자
Authorization : Bearer eyJqhdXic1J9.eyJ3Z.....HSajskzqodu					
{ "issueId": 1, "content": "댓글 내용", "createdAt": "2023-07-20T00:00:00Z" }					
Response Message					
HTTP	항목		반복여부	type	설명
	statusCode		N	Int	결과코드 200 이외에는 전부 에러
	message		N	Long	메시지
	data		N	Object	전송되는 데이터
{ "statusCode": 200, "message": "1번 이슈에 대한 댓글 생성 성공", "data": [] }					

댓글 조회

설명	해당 이슈에 대한 댓글을 조회한다.					
URL	/api/project/{projectId}/issues/{issueId}/memo					
Method	GET			Content-Type		
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 부여 받은 Access Token
Param	issueId		Y	N	Int	조회할 이슈 id
Authorization : Bearer eyJqhdXic1J9.eyJ3z.....HSajskzqodu						
Response Message						
HTTP	항목			반복여부	type	설명
	statusCode			N	Int	결과코드 200 이외에는 전부 에러
	message			N	String	메시지
	data			N	Object	전송되는 데이터
		id		N	Long	댓글 아이디
		issueId		N	Long	이슈 아이디
		memo_content		N	String	댓글 내용
		memberNickname		N	String	작성자 닉네임
		createdAt		N	Date	댓글 생성일자
		updatedAt		N	Date	댓글 수정일자
{ "statusCode": 200, "message": "1번 이슈의 모든 댓글 조회 성공", "data": [{ "id": 70, "issueId": 1, "memo_content": "박재석", "memberNickname": "박재석", "createdAt": "2023-07-18T06:43:55.449+00:00", "updatedAt": null }] }						

댓글 수정

설명	해당 이슈에 대한 댓글을 수정한다.				
URL	/api/project/{projectId}/issues/{issueId}/memo/{memoId}				
Method	PUT		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
param	projectId	Y	N	Long	이슈를 포함하는 프로젝트 id
	issueId	Y	N	Long	댓글을 포함하는 이슈 id
	memoId	Y	N	Long	댓글 id
body	issueId	Y	N	Long	댓글을 포함하는 이슈 id
	content	N	N	String	댓글 내용
	createAt	Y	N	Date	댓글 생성일자
Authorization : Bearer eyJqhdXic1J9.eyJ3z.....HSajskzqodu					
{ "issueId": 1, "content": "댓글 내용", "updatedAt" : "2023-07-20T00:00:00Z" }					
Response Message					
HTTP	항목	반복여부	type	설명	
	statusCode		N	Int	결과코드 200 이외에는 전부 에러
	message		N	Long	메시지
	data				전송되는 데이터
{ "statusCode": 200, "message": "1번 이슈에 대한 댓글 수정 성공", "data": [] }					

댓글 삭제

설명	해당 이슈에 대한 댓글을 삭제한다.				
URL	/api/project/{projectId}/issues/{issueId}/memo/{memoId}				
Method	DELETE		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
param	projectId	Y	N	Long	이슈를 포함하는 프로젝트 id
	issueId	Y	N	Long	댓글을 포함하는 이슈 id
Authorization : Bearer eyJqhdXic1J9.eyJ3z.....HSajskzqodu					
{ "id" = 1 }					
Response Message					
HTTP	항목		반복여부	type	설명
	statusCode		N	Int	결과코드 200 이외에는 전부 에러
	message		N	Long	메시지
	data		N	Object	전송되는 데이터
{ "statusCode": 200, "message": "1번 이슈에 대한 댓글 삭제 성공", "data": [] }					

4) 그룹 (생성은 회원가입 시에 함께 진행)

그룹 수정

설명	해당 그룹을 수정한다.				
URL	/api/group				
Method	PUT		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Body	code	Y	N	String	그룹 코드
	name	Y	N	String	그룹 이름
Authorization : Bearer eyJqhdXic1J9.eyJ3z.....HSajskzqodu					
{					
"name": "그룹 1",					
"code": "03919ba2-39aa-42d2-b2a3-3919777a99e3"					
}					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
{					
"statusCode": 200,					
"message": "그룹 수정 완료"					
}					

그룹 삭제

설명	해당 그룹을 삭제한다.				
URL	/api/group				
Method	DELETE		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Body	code	Y	N	String	그룹 코드
	name	N	N	String	그룹 이름
Authorization : Bearer eyJqhdlxic1J9.eyJ3z.....HSajskzqodu					
{ "name": "그룹 1", "code": "03919ba2-39aa-42d2-b2a3-3919777a99e3" }					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
{ "statusCode": 200, "message": "그룹 삭제 완료" }					

5) 프로젝트

프로젝트 생성

설명	새로운 프로젝트를 생성한다.				
URL	/api/project				
Method	POST		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Body	name	Y	N	String	프로젝트 이름
	description	Y	N	String	프로젝트 설명
	status	Y	N	String	프로젝트 상태
Authorization : Bearer eyJqhdXic1J9.eyJ3Z.....HSajskzqodu					
{ "name": "프로젝트 1", "description": "프로젝트 세부 설명", "status": "Stopped" }					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
{ "statusCode": 200, "message": "프로젝트 생성 성공" }					

프로젝트 정보 조회

설명	해당 프로젝트의 정보를 조회한다.					
URL	/api/project/{projectId}					
Method	GET		Content-Type			
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 부여 받은 Access Token
Param	projectId		Y	N	Long	프로젝트 id
Authorization : Bearer eyJqhdxlc1J9.eyJ3z.....HSajskzqodu						
Response Message						
HTTP	항목		반복여부	type	설명	
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러	
	message		N	String	정상결과/에러발생시의 메시지	
	data		N	Object	결과값 반환	
		id	N	Long	프로젝트 id	
		name	N	String	프로젝트 이름	
		description	N	String	프로젝트 설명	
		status	N	String	프로젝트 상태	
		createAt	N	String	프로젝트 생성 일자	
		groupName	N	String	프로젝트 그룹의 이름	
		groupCode	N	String	프로젝트 그룹의 코드	
<pre>{ "statusCode": 200, "message": "프로젝트 정보 조회 성공", "data": { "id": 1, "name": "aaaa", "description": "aaaa", "status": "Stopped", "createAt": "2023-07-13T02:07:16.773+00:00", "groupName": "aa", "groupCode": "ce0a2a6c-12da-44ce-8ba8-79eb9b9a7b8c" } }</pre>						

프로젝트 권한 조회

설명	해당 프로젝트의 내 권한을 확인합니다.				
URL	/api/project/{projectId}/role				
Method	GET		Content-Type		
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Param	projectId	Y	N	Long	프로젝트 id
Authorization : Bearer eyJqhdlxic1J9.eyJ3z.....HSajskzqodu					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
	data		N	String	PM
{ "statusCode": 200, "message": "프로젝트 권한 조회 완료", "data": "PM" }					

내가 속한 프로젝트 목록 조회

설명	내 그룹의 프로젝트 중 내가 팀원으로 속해 있는 프로젝트의 목록을 조회한다				
URL	/api/my-project				
Method	GET		Content-Type		
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Authorization : Bearer eyJqhdlxic1J9.eyJ3z.....HSajskzqodu					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
	data		N	Object	결과값 반환
		id	N	Long	프로젝트 id
		name	N	String	프로젝트 이름
		description	N	String	프로젝트 설명
		status	N	String	프로젝트 상태
		createAt	N	String	프로젝트 생성 일자
		updatedAt	N	String	프로젝트 업데이트 일자
		groupId	N	Long	프로젝트 그룹의 id
		pmname	N	String	프로젝트 PM 이름
		pmid	N	Long	프로젝트 PM id
{ "statusCode": 200, "message": "내가 속한 프로젝트 목록 조회 완료", "data": [{ "id": 10, "name": "프로젝트1", "description": "프로젝트 설명", "status": "In-progress", "createdAt": "2023-07-13T05:17:44.590+00:00", "updatedAt": "2023-07-13T05:17:44.590+00:00", "groupId": 1, "pmname": "박재석", "pmid": 1 }] }					

내가 속하지 않은 프로젝트 목록 조회

설명	내 그룹의 프로젝트 중 내가 팀원으로 속해 있지 않은 프로젝트의 목록을 조회한다				
URL	/api/other-project				
Method	GET		Content-Type		
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Authorization : Bearer eyJqhdlxic1J9.eyJ3z.....HSajskzqodu					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
	data		N	Object	결과값 반환
		id	N	Long	프로젝트 id
		name	N	String	프로젝트 이름
		description	N	String	프로젝트 설명
		status	N	String	프로젝트 상태
		createAt	N	String	프로젝트 생성 일자
		updatedAt	N	String	프로젝트 업데이트 일자
		groupId	N	Long	프로젝트 그룹의 id
		pmname	N	String	프로젝트 PM 이름
		pmid	N	Long	프로젝트 PM id
{ "statusCode": 200, "message": "내가 속하지 않은 프로젝트 목록 조회 완료", "data": [{ "id": 8, "name": "프로젝트", "description": "설명", "status": "Stopped", "createdAt": "2023-07-13T05:06:29.501+00:00", "updatedAt": "2023-07-13T05:06:29.501+00:00", "groupId": 1, "pmname": "b", "pmid": 2 }] }					

프로젝트 수정

설명	프로젝트의 정보를 수정한다.				
URL	/api/project				
Method	PUT		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Body	id	Y	N	Long	프로젝트 id
	name	N	N	String	프로젝트 이름
	description	N	N	String	프로젝트 설명
	status	N	N	String	프로젝트 상태
	role	N	N	String	프로젝트 권한
Authorization : Bearer eyJqhdlxic1J9.eyJ3z.....HSajskzqodu					
{ "id" : "1", "name": "aaa", "description": "vvv", "status": "Stopped" }					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
{ "statusCode": 200, "message": "프로젝트 수정 성공" }					

프로젝트 삭제

설명	선택한 프로젝트를 삭제한다.				
URL	/api/project				
Method	DELETE		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Body	id	Y	N	Long	프로젝트 id
Authorization : Bearer eyJqhdlxic1J9.eyJ3z.....HSajskzqodu					
{					
"id" : "1",					
}					
Response Message					
HTTP	항목	반복여부	type	설명	
Body	statusCode	N	Integer	결과코드 200 이외에는 전부 에러	
	message	N	String	정상결과/에러발생시의 메시지	
{					
"statusCode": 200,					
"message": "프로젝트 삭제 성공"					
}					

6) 릴리즈 노트

릴리즈 노트 생성

설명	새로운 릴리즈 노트를 생성한다.					
URL	/api/release					
Method	POST		Content-Type	application/json;charset=UTF-8		
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 부여 받은 Access Token
Body	version		Y	N	String	릴리즈 노트 버전
	status		Y	N	String	릴리즈 노트 상태
	percent		Y	N	String	릴리즈 노트 진행 상황
	releaseDate		Y	N	String	릴리즈 노트 생성 일자
	brief		Y	N	String	릴리즈 노트 요약
	description		Y	N	String	릴리즈 노트 설명
	projectId		Y	N	String	릴리즈 노트가 속한 프로젝트 ID
	member		Y	N	String	릴리즈 노트 작성자
	IssueList	Id	Y	Y	String	릴리즈 노트에 속한 이슈의 ID
Authorization : Bearer eyJqhdlxic1J9.eyJ3z.....HSajskzqodu						
{ "version": "0.0.0", "status": "backlog", "percent": 0, "releaseDate": "2023-07-21T00:14:54.757Z", "brief": "string", "description": "설명", "projectId": 1, "member": "박재석" "issueList": [{ "id": 1 }] }						
Response Message						
HTTP	항목		반복여부	type	설명	
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러	
	message		N	String	정상결과/에러발생시의 메시지	
{ "statusCode": 200, "message": "릴리즈 노트 생성 완료" }						

릴리즈 노트 조회

설명	선택한 프로젝트의 릴리즈노트 리스트를 조회한다.					
URL	/api/release/{releaseNoteld}					
Method	GET		Content-Type	application/json		
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 받은 access 토큰
Param	releaseNoteld		Y	N	Long	릴리즈노트 id
Authorization : Bearer [accessToken]						
Response Message						
HTTP	항목			반복여부	type	설명
Body	statusCode			N	Integer	응답 상태 코드
	message			N	String	상태 메시지
	data			N	Object	응답 데이터
		id		N	Long	릴리즈노트 id
		version		N	String	릴리즈노트 버전
		status		N	String	릴리즈노트 상태
		releaseDate		N	String	릴리즈노트 배포일자
		createdAt		N	String	릴리즈노트 생성일자
		member		N	Object	멤버 객체
			id	N	Long	멤버 id
			username	N	String	멤버 이름
			nickname	N	String	멤버 닉네임
{ "statusCode": 200, "message": "해당 릴리즈노트 조회 완료", "data": { "version": "1.0.1", "status": "Not released", "percent": 0.0, "releaseDate": "2023-08-01T00:00:00.000+00:00", "createdAt": "2023-07-20T01:20:57.470+00:00", "description": "릴리즈 노트 설명", "member": { "id": 1, "username": "박재석", "nickname": "박재석", } } }						

프로젝트에 해당하는 릴리즈 노트 목록 조회

설명	선택한 프로젝트의 릴리즈노트 리스트를 조회한다.				
URL	/api/project/{projectId}/release				
Method	GET		Content-Type		
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Param	projectId	Y	N	Long	프로젝트 id
Authorization : Bearer eyJqhdxlc1J9.eyJ3z.....HSajskzqodu					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
	data		N	Object	결과값 반환
		id	N	Long	릴리즈노트 id
		version	N	String	릴리즈노트 버전
		status	N	String	릴리즈노트 상태
		releaseDate	N	String	릴리즈노트 배포일자
		createdAt	N	String	릴리즈노트 생성일자
		member	N	Object	멤버 객체
			id	N	멤버 id
			username	N	멤버 이름
			nickname	N	멤버 닉네임
{ "statusCode": 200, "message": "릴리즈노트 목록 조회 완료", "data": [{ "id": 1, "version": "1.0.1", "status": "Not released", "releaseDate": "2023-08-01T00:00:00.000+00:00", "createdAt": "2023-07-20T01:20:57.470+00:00", "member": { "id": 1, "username": "박재석", "nickname": "박재석", } }] }					

릴리즈 노트 수정

설명	선택된 릴리즈 노트를 수정한다.					
URL	/api/release					
Method	PUT		Content-Type	application/json;charset=UTF-8		
Request Message						
HTTP	항목		필수여부	반복여부	type	설명
Header	authorization		Y	N	String	로그인시 부여 받은 Access Token
Body	version		Y	N	String	릴리즈 노트 버전
	status		Y	N	String	릴리즈 노트 상태
	percent		Y	N	String	릴리즈 노트 진행 상황
	releaseDate		Y	N	String	릴리즈 노트 생성 일자
	brief		Y	N	String	릴리즈 노트 요약
	description		Y	N	String	릴리즈 노트 설명
	projectId		Y	N	String	릴리즈 노트가 속한 프로젝트 ID
	member		Y	N	String	릴리즈 노트 작성자
	IssueList	Id	Y	Y	String	릴리즈 노트에 속한 이슈의 ID
Authorization : Bearer eyJqhdXic1J9.eyJ3Z.....HSajskzqodu						
{ "version": "0.0.0", "status": "backlog", "percent": 0, "releaseDate": "2023-07-21T00:14:54.757Z", "brief": "string", "description": "설명", "projectId": 1, "member": "박재석" "issueList": [{ "id": 1 }] }						
Response Message						
HTTP	항목		반복여부	type	설명	
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러	
	message		N	String	정상결과/에러발생시의 메시지	
{ "statusCode": 200, "message": "릴리즈 노트 수정 완료" }						

릴리즈 노트 삭제

설명	선택한 릴리즈 노트를 삭제한다.				
URL	/api/release				
Method	DELETE		Content-Type	application/json;charset=UTF-8	
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Body	id	Y	N	Long	릴리즈 노트 id
Authorization : Bearer eyJqhdlxic1J9.eyJ3z.....HSajskzqodu					
{					
"id" : "1",					
}					
Response Message					
HTTP	항목	반복여부	type	설명	
Body	statusCode	N	Integer	결과코드 200 이외에는 전부 에러	
	message	N	String	정상결과/에러발생시의 메시지	
{					
"statusCode": 200,					
"message": "릴리즈노트 삭제 성공"					
}					

7) 알림

알림 조회

설명	알림을 조회한다.				
URL	/api/project/{projectId}/notification				
Method	GET		Content-Type		
Request Message					
HTTP	항목	필수여부	반복여부	type	설명
Header	authorization	Y	N	String	로그인시 부여 받은 Access Token
Param	projectId	Y	N	Long	프로젝트 id
Authorization : Bearer eyJqhdXic1J9.eyJ3Z.....HSajskzqodu					
Response Message					
HTTP	항목		반복여부	type	설명
Body	statusCode		N	Integer	결과코드 200 이외에는 전부 에러
	message		N	String	정상결과/에러발생시의 메시지
	data		N		
		message	N	String	이슈에 대한 알림
		createdAt	N	String	알림 생성일자
<pre>{ "statusCode": 200, "message": "해당 프로젝트 알림 조회 성공" "data": ["message" : "# REA-1 이슈가 박재석에 의해 발행되었습니다.", "createdAt" : "2023-03-22"] }</pre>					
<div>● message(이슈에 대한 알림) 종류 이슈:</div> <div><div>- #{issueNum} 이슈가 000에 의해 발행되었습니다.</div><div>- #{issueNum} 이슈가 해결되었습니다.</div><div>- #{issueNum} 이슈가 삭제되었습니다.</div><div>- #{issueNum} 이슈가 000에게 할당되었습니다.</div><div>- #{issueNum} 이슈의 담당자가 000에서 000(으)로 변경되었습니다.</div></div> <div>릴리즈 노트:</div> <div><div>- ver.{version} 릴리즈 노트가 생성되었습니다.</div><div>- ver.{version} 릴리즈 노트가 수정되었습니다.</div><div>- ver.{version} 릴리즈 노트가 삭제되었습니다.</div></div> <div>멤버:</div> <div><div>- 프로젝트에 새로운 멤버({memberName})(이)가 참여하였습니다.</div><div>- 프로젝트에서 {memberName}가 탈퇴하였습니다.</div></div>					

13. QA 보고서

전체 QA는 '23. 8. 9. (수)에 진행하였으며, 페이지 별 오류 발생 및 수정 부분만 기술하였음.

페이지	URL	이슈
메인 홈	/home	Sidenav가 잠깐 나타났다가 사라지는 문제
프로젝트 목록	/home/manage_project	비로그인 상태로 url 직접 접속 시 Loading 화면만 출력되고 진행되지 않는 문제
로그인	/authentication/sign-in	틀린 아이디 또는 비밀번호를 입력하고 로그인 버튼 클릭 시 버튼만 비활성화되고 진행되지 않는 문제
마이페이지	/home/mypage	비로그인 상태로 url 직접 접속 시 흰 페이지만 출력되고 진행되지 않는 문제
		프로필 카드가 왼쪽으로 치우침
대시보드	/dashboard	이슈 표 세로 길이가 너무 짧음
멤버 정보	/members	멤버 수가 많을 시 페이지네이션 필요
멤버 프로젝트	/members-project	멤버 수가 많을 시 페이지네이션 필요
이슈 관리	/issuesearch	하위 이슈 선택 시 한 하위 이슈를 여러 번 클릭 하면 같은 이슈가 여러 개 추가되는 문제
		하위 이슈 여러 개를 선택해서 추가는 되지만 UI 상으로 여러 개가 선택 된 것인지 알아보기 힘들
		하위 이슈 추가 버튼 왼쪽으로 치우침
		사진 추가 Save file 버튼 시 스낵바 알림 필요
		이슈 세부 정보에서 담당자와 타입을 바꿔도 이슈 목록에서는 변경되지 않는 문제
		이슈 검색 필터에서 상태와 타입 담당자 필터를 여러 번 사용 시에 더이상 필터가 적용되지 않는 문제
칸반 보드	/issuemanagement	이슈 생성 시 담당자 지정 리스트에서 한 번에 표시되는 최대 인원 수 제한 및 오버플로우 시 스크롤 기능 필요
		제목을 입력하지 않아도 이슈가 생성되는 문제
		이슈 생성 시 생성일을 변경할 수 있는 문제
		이슈 생성 시 이미지가 추가되었는지 구분이 불가능한 문제

릴리즈 노트	/release	릴리즈노트 수정 시 버전 형식(x.x.x)를 지키지 않으면 생성이 되지 않으나 생성되었다는 알림이 뜨는 문제
		릴리즈 노트 생성 시 릴리즈 상태를 '예정'으로 생성하면 작성 날짜가 표시되지 않는 문제
		릴리즈 일자 입력을 캘린더 형식으로 변경 필요
		릴리즈 노트 삭제 시 정말 삭제할 것인지 재확인 필요
		릴리즈 노트 생성 시 버전에 음수를 입력해도 정상적으로 생성되는 문제
프로필	/profile	배지 사진에 레벨 글씨가 가려지는 문제
		본인 정보 수정 후 저장 버튼 클릭 시 모달창이 사라지지 않는 문제
		잔디 표 마우스 포인터 hover 시 날짜와 해결 이슈 개수 표시 위치가 잘려서 표시되는 문제
PM 페이지	/pm	프로젝트 설명에서 줄바꿈을 해도 모두 붙어서 표시되는 문제
		팀원이 4명 이상인 경우 가로로 스크롤 기능 필요

14. 인프라 보고서

1) Elastic Search 를 사용한 이유

목적: 키워드 검색

데이터가 적은 경우에는 MySQL 의 Full-Text-Search 는 Elasticsearch 와 거의 차이가 나지 않아서 데이터가 엄청 많지 않은 경우는 Elasticsearch 이 대안으로 충분히 사용할 수 있겠다는 생각이 들었다. 대용량 데이터나 전반적인 검색의 속도는 Elasticsearch 이 빨랐다.

- mysql 의 like 연산

일반적인 RDBMS 에서 텍스트로 데이터를 검색하기 위해서는 LIKE 연산, 즉 패턴 매칭으로 데이터를 탐색할 것이다. 이 방식은 테이블에 저장된 모든 데이터를 탐색하며 해당 컬럼의 데이터가 주어진 패턴과 일치하는지 여부를 따져보면서, 결과를 필터링할 것이다.

보통 mysql 에서 텍스트 문자열을 검색시 like 나 instr 연산자를 사용해 원하는 텍스트를 필터링하여 조회할 수 있다. 그러나 검색할 텍스트의 길이가 긴 경우 성능이 떨어지게 될 것이다. 만약 10 년치 기사에서 검색한다면 과부하가 발생하고 응답시간도 길어진다. 이것을 전체 텍스트 검색 기능이 해결해 준다. 전체 텍스트 검색은 첫 글자 뿐 아니라 중간의 단어나 문장으로도 인덱스를 생성해 주기 때문에, 전체 텍스트 인덱스를 통해 순식간에 검색 결과를 얻을 수 있다.

전체 텍스트 인덱스는 신문기사와 같이, 텍스트로 이루어진 문자열 데이터의 내용을 가지고 생성한 특수한 인덱스 종류이다. 내용이 긴 텍스트를 따로 인덱싱하여 관리할 수 있다.

- Mysql 의 Full text search

전체 텍스트 검색은 긴 문자의 텍스트 데이터를 빠르게 검색하기 위한 MySQL 의 부가적인 기능이다. 전체 텍스트도 결국 인덱스의 종류의 한가지이다. 하지만 대용량 데이터에서는 full text search 의 성능이 떨어질 수 있다.

검색 정확성 부족: Full Text Search 는 텍스트 데이터를 단어로 나누어 검색하는데, 이 때 불용어(stop words)나 특수 문자 등은 무시되기도 한다. 이로 인해 정확한 검색 결과가 제공되지 않을 수 있다.

불용어(stop words)는 텍스트 데이터에서 자주 등장하지만 일반적으로 검색이나 분석을 수행할 때 무시되는 단어나 문자들을 말한다. 이러한 단어들은 문장에서는 자주 등장하지만 검색어로서는 의미를 가지지 않거나 검색 결과를 좀 더 정확하게 만들기 위해 제외된다.

참고) 불용어: 예를 들어, "the", "and", "is", "in" 등의 단어들.

- Elasticsearch

반면 엘라스틱 서치의 역 인덱스 방식은 조금 다르다. 일단 텍스트를 여러개의 키워드로 쪼갬다. 그리고 이 쪼갬 키워드는 자기 자신이 등장한 도큐먼트를 가리킨다. 예를 들어 'hudi' 라는 단어가 5 번째, 6,520 번째, 10,000 번째 도큐먼트에서 등장한다고 가정하자. 일반적인 RDBMS 는 'LIKE %hudi%' 로 검색했을 때 10,000 번째 행까지 탐색을 하겠지만, 역 인덱스 구조에서는 'hudi' 라는 키워드가 가리키는 도큐먼트가 무엇인지 확인만 하면 된다. 아래 그림을 보자.

Document 1 ~ 3 의 내용을 일반적인 RDBMS 에 저장하고, 'LIKE' 연산을 통해 패턴 매칭하여 데이터를 탐색한다고 해보자. 저 중 'search' 라는 단어가 포함된 문서를 검색해보자. Inverted Index 에서 바로 'search' 행을 찾아보면 Document 3을 가리키는 것을 확인할 수 있다. 마치 해시 테이블을 사용하는 것 처럼 시간 복잡도가 거의 $O(1)$ 에 수렴한다.

2) RabbitMQ를 사용한 이유

메시지 큐의 이점

1. 비동기(Asynchronous)

메시지 큐는 생산된 메시지의 저장, 전송에 대해 동기화 처리를 진행하지 않고, 큐에 넣어두기 때문에 나중에 처리할 수 있다. 여기서 기존 동기화 방식은 많은 메시지를 전송할 경우 병목 현상이 생길 수도 있고, 뒤에 들어오는 요청에 대한 응답이 지연될 것이다.

2. 낮은 결합도(Decoupling)

생산자 서비스와 소비자 서비스가 독립적으로 행하면서 서비스간 결합도가 낮아진다.

3. 확장성(Scalable)

생산자 서비스 혹은 소비자 서비스를 원하는 대로 확장할 수 있기 때문에 확장성이 좋다.

4. 탄력성(Resilience)

소비자 서비스가 다운되더라도 어플리케이션이 중단되는 것이 아니라 메시지는 메시지 큐에 남아있다. 소비자 서비스가 다시 시작될 때마다 추가설정이나 작업을 수행하지 않아도 메시지 처리를 시작할 수 있다.

5. 보장성(Guarantees)

메시지 큐는 큐에 보관하는 모든 메시지가 결국 소비자 서비스에게 전달된다는 일반적인 보장을 제공한다.

- 카프카 vs rabbitmq

카프카가 더 복잡한 아키텍처를 사용하기에 처리량이 높은, 대용량 스트림 이벤트를 처리할 수 있다.

Kafka 는 메시지를 스트리밍할 때 지연 시간이 매우 짧기 때문에 스트리밍 데이터를 실시간으로 분석하는 데 적합합니다. 예를 들어 Kafka 를 분산 모니터링 서비스로 사용하여 온라인 트랜잭션 처리 알람을 실시간으로 생성할 수 있다.

따라서 Kafka 는 수신된 데이터를 다시 분석해야 하는 애플리케이션에 적합하다. 보존 기간 내에 스트리밍 데이터를 여러 번 처리할 수도 있고 로그 파일을 수집하여 분석할 수도 있다. 반면 사용된 메시지는 삭제되기 때문에 RabbitMQ 를 사용하여 로그를 집계하기는 더 어렵다.

RabbitMQ 는 푸시 모델을 적용한다. 즉, 생산자는 클라이언트 애플리케이션이 메시지를 사용했는지 여부를 알 수 있다. 따라서 데이터를 교환하고 분석할 때 특정 순서 및 전달 보장을 준수해야 하는 애플리케이션에 적합하다.

보통 MQTT 및 STOMP 와 같은 레거시 프로토콜과 호환되어야 하는 클라이언트 애플리케이션에 RabbitMQ 를 사용한다.

- RabbitMQ 의 메시지 손실 최소화 방법

1. RabbitMQ 노드를 클러스터로 사용. 하나의 노드가 죽어도 다른 노드가 커버할 수 있도록 설정하기.

2. Queue 의 durability (durable 속성)를 true 로 설정해두기.

노드가 커졌다가 다시 켜졌을 때 라우팅 테이블이 초기화되지 않도록, 브로커 노드가 재가동되어도 이전의 정보들을 가지고 복원이 될 수 있도록 설정하는 것.

3. 큐 미러링.

한 클러스터 내에 존재하는 큐 노드는 서로 동일한 메시지를 가지지 않는다. 따라서 A 라는 메시지를 받은 노드가 죽었다면, 그 노드가 재가동되기 전까지는 A 메시지를 받을 수 없다. 이를 방지하기 위해 큐 미러링을 사용한다. 큐 미러링이란 한 클러스터 내의 노드 각자가 가지는 메시지가 서로 동기화 되도록 설정하는 것이다. 큐 미러링을 사용했을 때 메시지 신뢰성은 높아지겠지만 이는 전체적인 성능 저하를 일으킬 수 있다. 따라서 상황에 맞게, 중요도에 따라 적절히 큐의 개수를 조정해야 한다.

4. 큐의 마스터, 슬레이브를 구성하여 서버를 구성한다. 마스터 큐가 먼저 메시지를 받아서 슬레이브 큐에 전달해주는 방식이다. 이 때에 고려해야 하는 것은 '마스터 슬레이브가 죽었을 때 마스터를 담당할 슬레이브 노드를 어떤 순서로 정할 것인지' 이다.

3) Docker Swarm을 사용한 이유

Docker Swarm 은 Docker 에서 제공하는 컨테이너 오케스트레이션 툴이다. 사용하는 컨테이너의 수가 많을 때 관리 및 배포에 효율성을 제공하며, 워커 노드에 컨테이너 자동 배치, 로드 밸런싱, 장애 복구 등의 기능을 제공한다. 대표적인 컨테이너 오케스트레이션 툴로 도커 스웜과 쿠버네티스가 있다.

- Docker Swarm vs. Kubernetes

Docker Swarm

장점

1. Docker Swarm 은 Docker Engine 에 이미 포함되어 있는 기능이기 때문에 Docker CLI 를 통해 (상대적으로) 손쉽게 다룰 수 있다.
2. Swarm 만의 API 가 있다.
3. Docker Compose 처럼 Docker 의 다른 도구들과 부드럽게 호환된다.
4. Docker Container 에서 동작하는 도구, 서비스, 소프트웨어는 Docker Swarm 위에서도 잘 동작한다.
5. 환경 설정이 비교적 쉬워서 간단하게 사용할 수 있다.
6. Container 배포 시 필터링 및 스케줄 기능을 제공한다.

단점

1. 커스터마이징과 확장에 있어 한계가 있다.
2. Kubernetes 보다 기능이 적다.
3. DevOps 파이프라인에서 Dev-Test-Prod 업무를 분리하기가 쉽지 않다.

Kubernetes

장점

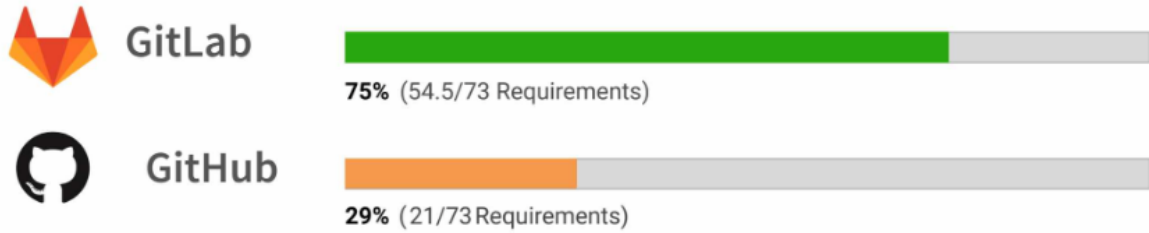
1. Open Source Community 가 매우 활성화되어 있어 질문을 하거나 정보를 얻을 때 도움을 많이 받을 수 있다.
2. 연중 컨퍼런스 참석자가 2 배 이상 증가할 정도로 커뮤니티가 빠르게 증가한다.
3. 대부분의 OS 에서 Google, IBM 과 같은 큰 기업들이 산업 현장에서 이미 많은 검증을 하였다.
4. 시장에서 가장 많이 채택되고 있다.
5. AWS, Azure, GCP, IBM Cloud 등 Public Cloud 나 on-premises 에서 사용 가능하다.
6. 서비스 탐색, 로드밸런싱, 자가복구 등 주요 기능들을 가지고 있다.
7. Cluster 상태 관리를 잘 보장하고 있고 잘 연합된 API 세트를 가지고 있다.

단점

1. 초기 learning curve 가 존재한다.
2. Open Source 커뮤니티에서 업데이트가 빈번하기 때문에 패치 시 유의해야 한다.
3. 간단한 기능에 빈번하지 않은 배포를 하는 개별 개발자가 사용하기에는 너무 무겁다.
4. 접근제어, 식별, 관리 및 보안 등 프로그램을 잘 배포하고 유지하기 위해 종종 kubectl CLI 와 같은 추가적인 도구들이나, CI/CD 툴, 기타 DevOps 툴 등이 필요하다.

Docker Swarm 과 Kubernetes 를 종합적으로 판단해 보았을 때, 작고 가벼운 프로젝트에서는 손쉽게 사용할 수 있는 Docker Swarm 을 사용하고, 프로젝트가 커지거나 엄중한 관리가 필요한 시점에서는 Kubernetes 를 사용하는 게 좋은 전략이라 생각한다.

4) GitLab을 사용한 이유



- Gitlab 이란?

소프트웨어 프로젝트를 무료로 호스팅한다.

Git 리포지토리를 관리하기 위한 플랫폼이다.

무료 공개 및 비공개 리포지토리, 문제 추적 및 위키를 제공한다.

git 위에있는 사용자 친화적인 웹 인터페이스 계층으로 Git 작업 속도를 높일 수 있다.

프로젝트 관리를 위한 자체 CI (Continuous Integration) 시스템을 제공하고 GitLab 의 다른 기능과 함께 사용자 인터페이스를 제공한다.

장점

- GitLab 은 코드가있는 서버에서 사용자가 찾을 수 있도록 Gitlab Community Edition 버전을 제공한다.
- 무제한의 개인 및 공용 리포지토리를 무료로 제공한다.

단점

- 리포지토리를 PUSH PULL 기능이 Github 만큼 빠르지는 않다.
- 한 페이지에서 다른페이지로 전환하는 동안 GitLab 인터페이스 시간이 걸린다.

- 선택 이유

Gitlab 은 사용자 수를 제한하지 않는 무료 개인 및 공용 레포지토리를 제공하고, 여러 레포지토리를 호스팅 및 많은 사람들과 함께 작업하기에 유용하다.

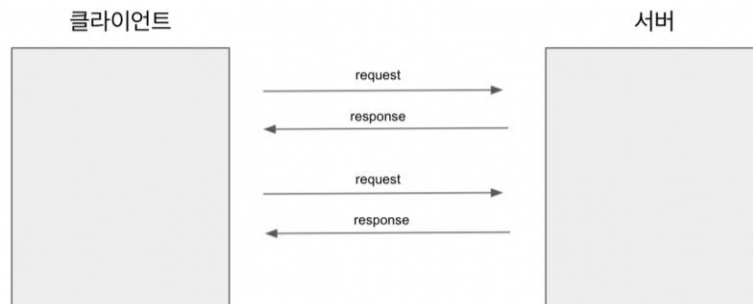
또한 기업에서는 설치형 Gitlab 을 쓴다고 하는데, 실무에 들어가기 전에 회사에서 이용하는 방식을 경험해보기 위해 선택했다.

5) SSE를 사용한 이유

실시간 웹 애플리케이션 개발 시 사용되는 몇 가지 방법이 있다.

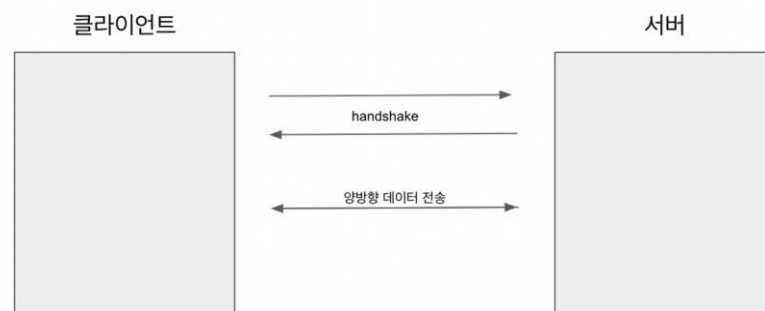
- polling(client pull)

클라이언트가 일정한 주기로 서버에 업데이트 요청을 보내는 방법. 지속적인 HTTP 요청이 발생하기 때문에 리소스 낭비가 발생한다.



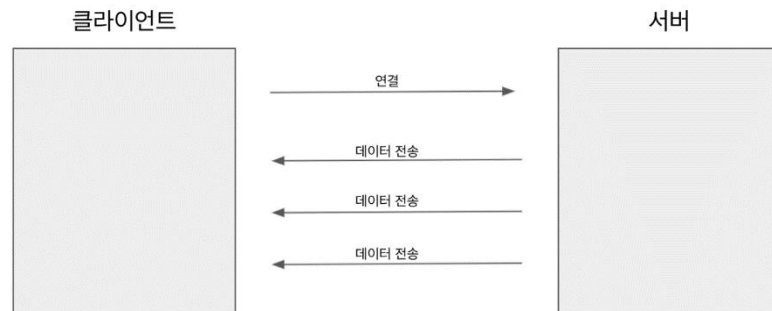
- WebSocket(server push)

실시간 양방향 통신을 위한 스펙으로 서버와 브라우저가 지속적으로 연결된 TCP 라인을 통해 실시간으로 데이터를 주고받을 수 있도록 하는 HTML5 사양이다. 연결지향 양방향 전이중 통신이 가능하며 채팅, 게임, 주식 차트 등에 사용된다. polling 은 주기적으로 HTTP 요청을 수행하지만, websocket 은 연결을 유지하여 서버와 클라이언트 간 양방향 통신이 가능하다.₩



- SSE(server push)

SSE 는 서버의 Event 를 stream 하는 기술이다. 일반적 HTTP 통신이 요청에 따른 데이터를 제공한 뒤 연결을 끊는 것과는 달리, 초기 서버 이벤트를 구독해 놓으면, 서버에서는 지정된 이벤트가 발생 할 때마다 Client 로 데이터를 보낼 수 있다.



SSE 는 서버의 데이터를 실시간, 지속적으로 클라이언트에 보내는 기술이다. 위의 그림처럼 클라이언트에서 처음 HTTP 연결을 맺고 나면 서버는 클라이언트로 계속하여 데이터를 전송할 수 있다.

일반적으로 HTTP 요청은 하나의 [요청 - 응답] 과정을 거치고 연결을 종료한다. 하지만 파일 전송과 같이 연결 상태를 유지하고 계속 데이터를 보내는 경우도 있다. SSE 는 이와 같이 한 번 연결 후 서버에서 클라이언트로 데이터를 계속해서 보낼 수 있다.

특징

- websocket 과 달리 별도의 프로토콜을 사용하지 않고 HTTP 프로토콜만으로 사용이 가능하며 훨씬 가볍다.
- 접속에 문제가 있으면 자동으로 재연결을 시도한다.
- 최대 동시 접속 수는 HTTP/1.1 의 경우 브라우저 당 6 개이며, HTTP/2 는 100 개까지 가능하다.
- IE 를 제외한 브라우저에서 지원된다.(Polyfills 을 사용하면 가능하다고 한다.)
- 이벤트 데이터는 UTF-8 인코딩된 문자열만 지원한다. 일반적으로 JSON 으로 파싱하여 전송한다.
- 클라이언트에서 페이지를 닫아도 서버에서 감지하기가 어렵다.

3 가지 방법 중 **SSE** 를 이용하여 실시간 알림 기능을 구현하기로 결정했다. polling 은 지속적인 요청을 보내야 하므로 리소스 낭비가 심할 것 같았고, 실시간 알림 같은 경우는 서버에서 클라이언트 방향으로만 데이터를 보내면 되기 때문에 websocket 처럼 양방향 통신은 필요 없었다. 따라서 웹 소켓에 비해 가볍고 서버 -> 클라이언트 방향을 지원하는 SSE 를 선택했다.

15. 성능 테스트 결과

프론트엔드 성능 테스트

이 보고서는 릴리즈 노트 플랫폼 releasy 웹 애플리케이션의 프론트엔드 성능을 평가하기 위한 테스트 결과를 요약하여 제공한다. 성능 최적화는 사용자 경험을 향상시키는 중요한 요소이며, 이를 통해 사이트의 전환율, 사용자 만족도 및 전반적인 효율성을 향상시킬 수 있다.

테스트 목표

1. 속도 개선: 웹 페이지 로딩 속도를 개선하여 사용자가 빠르게 콘텐츠를 볼 수 있도록 한다. Lighthouse 는 페이지의 로딩 시간과 관련된 여러 요소를 분석하여 지연되는 요소를 식별하고 최적화 방법을 제안한다.
2. 사용자 경험 향상: 웹 페이지의 사용자 경험을 높이기 위해 Lighthouse 는 페이지의 시각적 안정성, 상호작용 가능성, 페이지 내용의 변동성 등을 평가합니다. 이를 통해 사용자가 웹 페이지에서 더 나은 경험을 얻을 수 있도록 돕는다.
3. 접근성 개선: 장애를 가진 사용자들을 위한 웹 페이지의 접근성을 향상 시키는 것도 중요한 목표이다. Lighthouse 는 웹 콘텐츠 접근성 지침 (WCAG) 준수 여부를 확인하고 접근성 관련 문제를 식별하여 개선 방안을 제시한다.
4. SEO (검색 엔진 최적화) 향상: 검색 엔진에서의 노출을 개선하기 위해 페이지의 SEO 상태를 평가한다. 또한 웹 페이지의 메타데이터, 구조화된 데이터, 페이지 속도 등이 검색 엔진 최적화에 어떻게 영향을 미치는지를 분석한다.
5. 성능 모니터링: Lighthouse 는 웹 페이지의 성능을 측정하고 추적하는 도구로 사용될 수 있다. 정기적으로 테스트를 수행하여 성능의 변화를 모니터링하고 문제가 발생할 경우 조치를 취할 수 있다.

테스트 환경

- 클라우드: Kakao I Cloud
- 애플리케이션 버전: 1.2.0
- 서버 스펙: CPU 2 코어, RAM 8GiB
- 테스트 툴: lighthouse
- 테스트 제품: LG 18 년도 울트라 노트북

테스트 결과

FCP (First Contentful Paint): 첫 요소가 로드될 때까지 걸리는 시간

FMP (First Meaningful Paint): 사용자에게 의미 있는 첫 요소가 로드될 때까지 걸리는 시간

LCP (Largest Contentful Paint): 주요 콘텐츠가 로드될 때까지 걸리는 시간

FID (First Input Delay):

사용자의 행동에 대해 실제로 이벤트 핸들러가 반응하기까지 걸리는 시간을 의미한다. 사용자가 이벤트를 발생시킬 때 메인 스레드에서 다른 작업이 진행 중이라면 그 이벤트를 처리할 수 있는 시점은 해당 작업이 끝난 이후가 된다. 이처럼 사용자의 이벤트를 받고 실제로 처리할 수 있게 될 때까지 걸리는 시간을 FID 라고 한다. 이 시간이 짧을수록 사용자는 입력에 대한 반응이 빠르게 온다고 느끼게 될 것이다. (Good: ~ 100ms, Need Improvement: 100ms ~ 300ms, Poor: 300ms ~)

CLS (Cumulative Layout Shift):

시각적인 안정성을 측정하는 데 사용되는 기준이다. 이는 시작 위치에서 레이아웃이 얼마나 변화하는지에 대해 측정한다. 다만, 이런 레이아웃이 글 작성 후 작성 완료 버튼을 누른 이후의 화면 변화와 같이 사용자가 충분히 예측할 수 있는 것이라면 변경된 정도가 커져도 성능에 영향을 미친다고 보지 않는다. (Good: ~ 0.1, Need Improvement: 0.1 ~ 0.25, Poor: 0.25 ~)

Lighthouse 는 Performance, Accessibility, BestPractices, SEO 를 점수화 해서 보고서로 작성해준다.

Performance : First Contentful Paint(최초 콘텐츠가 포함된 페인트), Largest Contentful Paint(가장 큰 콘텐츠가 포함된 페인트), Total Blocking Time(총 차단 시간), Cumulative Layout Shift(누적 레이아웃 이동) , Speed Index(속도 지수)를 통해서 도출

Accessibility: 웹 애플리케이션의 접근성을 검사

Best Practices : 웹 페이지가 웹에 대한 표준 모범 사례를 따르고 있는지 확인. 웹 애플리케이션을 가동할 때 콘솔에 오류가 출력 되진 않는지, 더는 사용하지 않는 API 를 호출하고 있지 않은지, HTTPS 를 통해 해당 페이지에 접근할 수 있는지 확인

SEO : 웹 페이지가 검색 엔진에 대해 최적화된 순위 결과를 가지고 있는지 확인: 콘텐츠를 읽는 데에 무리가 없는 글꼴 크기를 사용하는지, 웹 페이지의 robots.txt 파일이 유효한지, 올바른 상태 코드를 사용하는지 등

랜딩 페이지(/home)

23. 8. 17. 오후 5:33

about:blank



http://61.109.214.103/home



Performance



Accessibility



Best Practices



SEO



PWA



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0-49

50-89

90-100



METRICS

Expand view

▲ First Contentful Paint

3.1 s

Total Blocking Time

170 ms

▲ Speed Index

9.6 s

▲ Largest Contentful Paint

4.0 s

Cumulative Layout Shift

0

23. 8. 17. 오후 5:02

about:blank

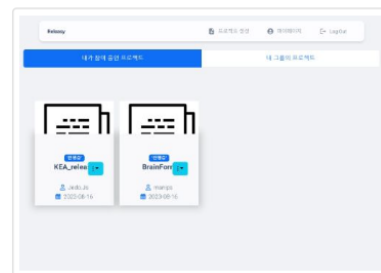


<http://61.109.214.103/home/manage-project>



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)



METRICS

Expand view

First Contentful Paint

0.8 s

▲ Largest Contentful Paint

3.3 s

Total Blocking Time

110 ms

Cumulative Layout Shift

0

Speed Index

0.8 s

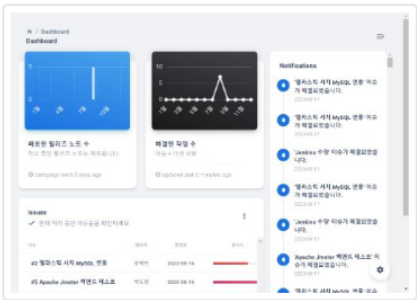
프로젝트 별 대시보드(/dashboard)



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0–49 50–89 90–100



METRICS

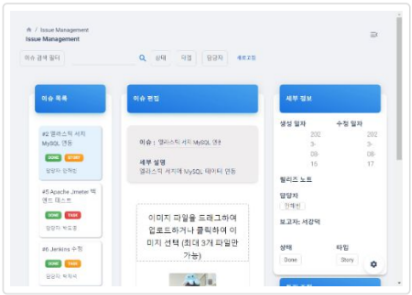
Expand view

First Contentful Paint 0.9 s	▲ Largest Contentful Paint 3.6 s
▲ Total Blocking Time 890 ms	Cumulative Layout Shift 0.077
Speed Index 1.9 s	



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)



METRICS

[Expand view](#)

First Contentful Paint

0.8 s

▲ Largest Contentful Paint

3.6 s

Total Blocking Time

220 ms

Cumulative Layout Shift

0.034

Speed Index

2.0 s

릴리즈 노트 페이지(/release)



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

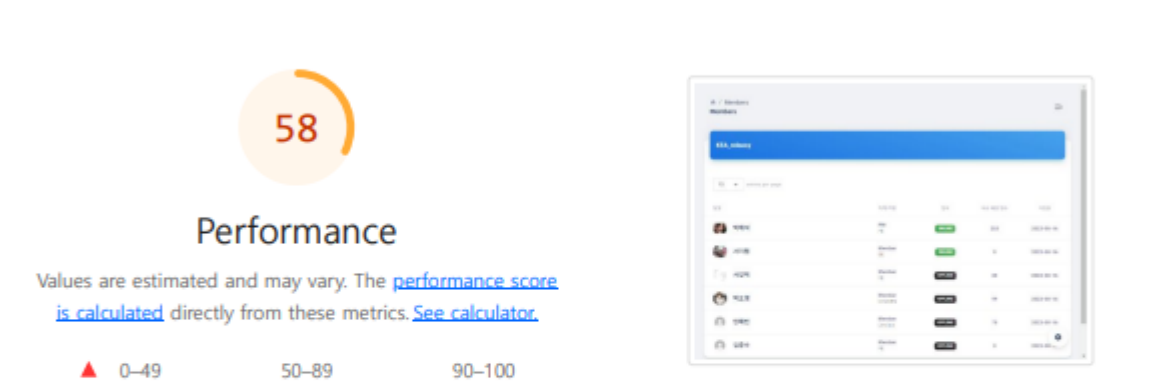
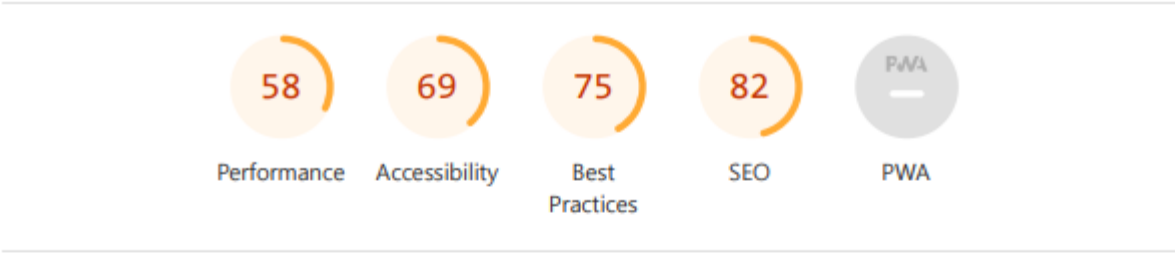
▲ 0–49 50–89 90–100

버전	2021.07.15	2021.07.15	2021.07.15	2021.07.15	2021.07.15
2.0.0	2023-08-24	2023-08-24	2023-08-24	2023-08-24	Not released
1.2.0	2023-08-17	2023-08-17	2023-08-17	2023-08-17	Not released
1.1.1	2023-08-16	2023-08-16	2023-08-16	2023-08-16	Released
1.1.0	2023-08-24	2023-08-24	2023-08-24	2023-08-24	Released
1.0.0	2023-08-16	2023-08-16	2023-08-16	2023-08-16	Released

METRICS

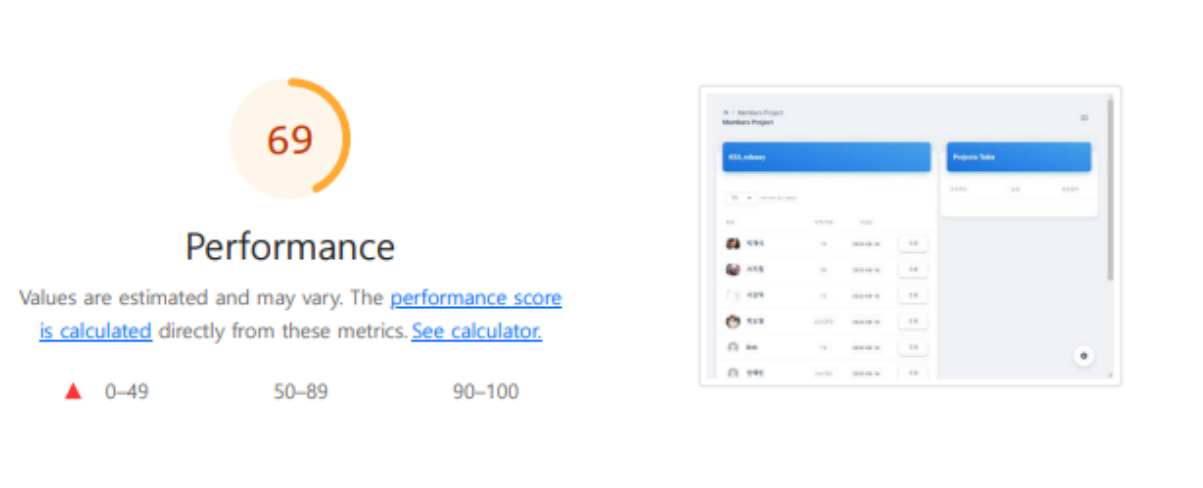
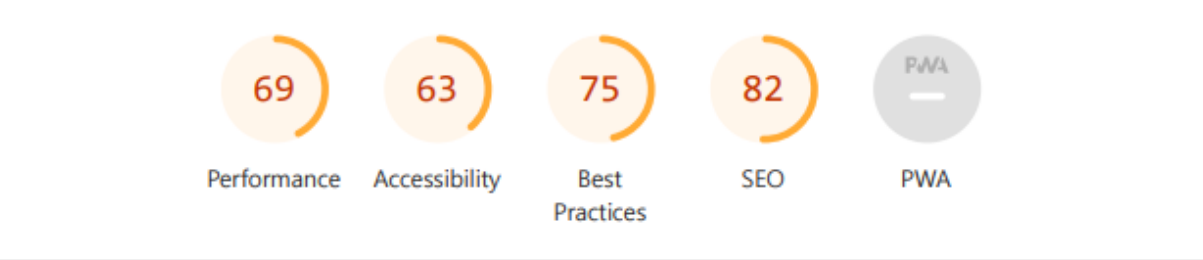
Expand view

First Contentful Paint 0.8 s	▲ Largest Contentful Paint 3.0 s
Total Blocking Time 130 ms	Cumulative Layout Shift 0.097
Speed Index 2.0 s	



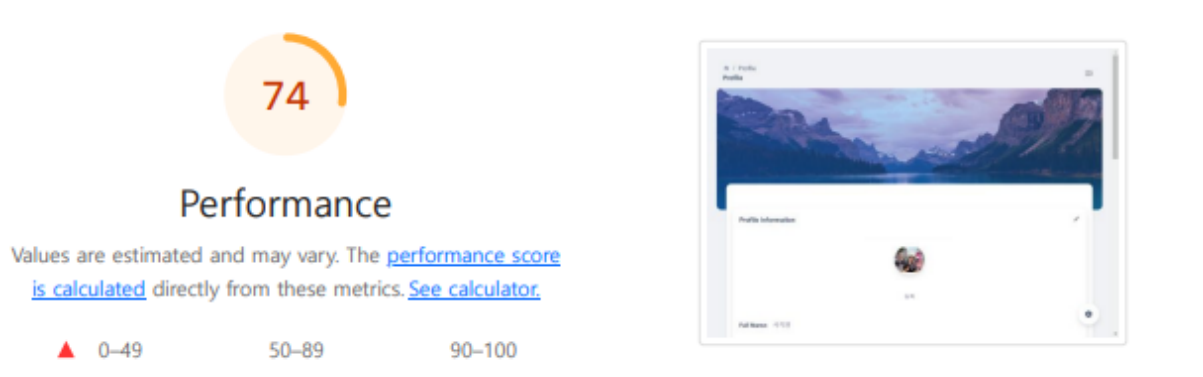
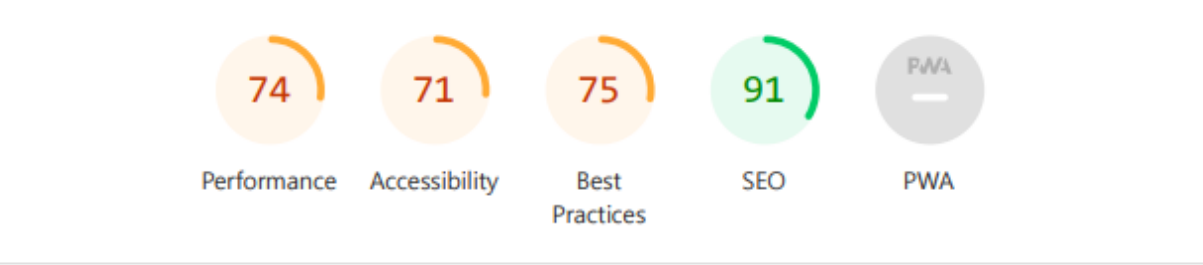
METRICS		Expand view
First Contentful Paint	0.8 s	▲ Largest Contentful Paint 3.4 s
Total Blocking Time	150 ms	▲ Cumulative Layout Shift 0.298
Speed Index	2.1 s	

그룹 멤버들의 참여 중인 프로젝트 조회 페이지(/members-project)



METRICS		Expand view
First Contentful Paint	1.0 s	▲ Largest Contentful Paint 3.2 s
Total Blocking Time	200 ms	Cumulative Layout Shift 0.069
Speed Index	2.3 s	

프로필 페이지(/profile)



METRICS		Expand view
First Contentful Paint	0.8 s	▲ Largest Contentful Paint 3.7 s
Total Blocking Time	160 ms	Cumulative Layout Shift 0.073
Speed Index	1.5 s	

프로젝트 관리 페이지: PM 페이지(/pm)



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0-49 50-89 90-100



METRICS

[Expand view](#)

First Contentful Paint

0.8 s

▲ Largest Contentful Paint

3.1 s

Total Blocking Time

140 ms

Cumulative Layout Shift

0.074

Speed Index

1.9 s

전반적인 결과 분석

공통적으로 나온 점수 하락 이유들 : 이미지 요소에 명시적 너비와 높이가 없음, 웹 폰트 로드 중에 텍스트가 계속 보이도록 함, 네트워크 페이 로드 방지, 버튼에 접근 가능한 이름, 양식요소 라벨, 링크 구별 이름 등도 접근성 점수 하락의 원인이 되었다. Best Practices 는 HTTPS 를 사용하지 않는 것의 문제와 Exception 로그들의 처리를 제대로 하지 못해서 브라우저 콘솔에 에러가 많이 뜬 점이 점수 하락 원인이었다.

1. 이미지 요소에 명시적 너비와 높이가 없다.

정적 이미지 , 오브젝트 스토리지에서 불러오는 이미지 파일의 명시적 너비와 높이가 존재하지 않는다. 따라서 이미지가 로드 되면서 차지하는 공간이 변하면서 페이지 다른 주변 요소들이 재배치되어 렌더링 시간 뿐 아니라 의도치 않은 사용자 경험을 야기할 수 있다.

2. 웹 폰트 로드 중에 텍스트가 계속 보이도록 한다.

웹 폰트는 웹 페이지의 디자인과 레이아웃을 개선하는 데 도움이 되지만, 폰트 파일의 크기가 크거나 폰트 파일을 로드하기 위해 추가적인 네트워크 요청이 필요할 경우 성능 문제를 야기할 수 있다. 특히 폰트 파일이 크고 네트워크 속도가 느린 상황에서는 웹 폰트 로드가 페이지 로딩 시간을 늘리고 사용자 경험을 저해할 수 있다.

3. 에러 로그들과 불필요한 로그들이 콘솔에 출력되어 Best practices 에 악영향을 미치고 있다.

4. HTTP 사용 및 Exception 처리가 부족하다.

5. 네트워크 페이로드가 크다.

네트워크 페이로드가 커지면 사용자는 더 많은 셀룰러 데이터를 사용해야 하며, 로드 시간이 길어질 수 있다. 불필요한 공백 및 코드를 제거하고, 이미지 포맷을 JPEG/PNG 대신, WebP 를 사용한다. JPEG 이미지의 압축 수준을 85 로 설정하는 등의 관리 방법이 있다.

6. 캐시의 활용성이 낮다.

정적인 자원들에 대해서 캐시를 활용하면 조금 더 렌더링 속도와 성능에 더 좋은 영향을 미칠 수 있다.

7. 최적화되지 않고 사용하지 않는 JS 와 CSS 를 사용한다.

코드를 컴포넌트화 시키지 않아 반복되는 코드로 인해 파일 크기가 커지고 있다. 따라서 사용하지 않는 자바스크립트를 줄이고 스크립트가 필요할 때까지 로딩을 지연시켜 네트워크 활동에 소비되는 바이트를 줄여야 한다.

백엔드 성능 테스트

이 보고서는 Releasy 웹 애플리케이션의 성능 테스트 결과에 대한 보고서이다. 해당 테스트는 부하와 응답 시간 등의 성능 지표를 측정하여 애플리케이션의 성능을 평가하였다.

테스트 목표

- 시스템 부하에 따른 성능 및 응답 시간 파악
- 병목 현상 확인 및 개선할 부분 도출

테스트 환경

- 클라우드: Kakao I Cloud
- 애플리케이션 버전: 1.2.0
- 서버 스펙: CPU 2 코어, RAM 8GiB
- 데이터베이스: MySQL 8.0.32
- 테스트 툴: JMeter 5.6.2

테스트 시나리오

1. 기능 테스트

- 시나리오 1: 메인 페이지와 상위 페이지(프로젝트 관련 페이지) 접속 (테스트 케이스 수: 5)
- 시나리오 2: 하위의 모든 페이지 접속 (테스트 케이스 수: 7)

2. 비기능 테스트

- 시나리오 3: 동시 접속자 수별 응답시간 측정

동시 접속자 수 조정 후 테스트: 100 명, 500 명, 1000 명

테스트 결과

1. 기능 테스트

- 시나리오 1: 메인 페이지와 상위 페이지(프로젝트 관련 페이지) 접속 - 테스트 케이스 수: 5

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
나의 프로젝트 조회	10	536	0	556	12.09	0.00%	18.0/sec	263.50	6.29	15029.0
내가 참여 중이지 않은 프로...	10	68	0	73	4.04	0.00%	90.1/sec	70.38	31.85	800.0
프로젝트 정보 수정	10	69	0	76	5.22	0.00%	84.7/sec	33.10	39.39	400.0
프로젝트 생성	10	103	0	112	5.94	0.00%	64.1/sec	25.04	29.55	400.0
TOTAL	40	194	0	556	197.93	0.00%	50.3/sec	204.27	20.50	4157.2

시나리오 1-1. 분석 및 해석

- "나의 프로젝트 조회" 시나리오에서 평균 응답 시간이 536ms 로 높게 나왔다. 다소 느린 응답 시간을 가지고 있으며, 데이터의 크기 및 처리 과정의 복잡성을 고려해야 한다.

- 내가 참여 중이지 않은 프로젝트 조회: 평균 응답 시간은 68ms 로, 적절한 수준의 성능을 나타낸다. 빠른 응답 시간을 보이며, 해당 기능은 성능 면에서 문제 없이 동작하고 있는 것으로 분석된다.

- 프로젝트 정보 수정: 평균 응답 시간은 69ms 로, 다른 시나리오에 비해 적절한 성능을 보입니다. 프로젝트 정보 수정은 단순하고 가벼운 작업이므로, 빠른 응답 시간을 유지하고 있는 것으로 보인다.

- "프로젝트 생성" 시나리오의 평균 응답 시간은 103ms 로 중간 정도의 수준을 보이고 있다.

시나리오 1-2. 성능 개선 사항 및 결론

- "나의 프로젝트 조회" 시나리오에서 높은 응답 시간을 보이므로 데이터베이스 쿼리 최적화나 프로세스 개선을 통해 성능을 향상시킬 필요가 있다.

- 다른 시나리오에서는 만족스러운 성능을 보이고 있다.

시나리오 1-3. 결과 활용 및 추후 계획

- "나의 프로젝트 조회" 시나리오의 성능 개선을 위해 데이터베이스 인덱스를 검토하고, 데이터 조회 방식을 최적화하는 작업을 계획하고 있다.

- 시나리오 2: 하위의 모든 페이지 접속 (테스트 케이스 수: 7)

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
모든 이슈 조회	1	793	0	793	0.00	0.00%	1.3/sec	8.25	0.45	6697.0
알림 내역 조회	1	84	0	84	0.00	0.00%	11.9/sec	269.47	4.38	23179.0
프로젝트의 릴리즈노트 조회	1	39	0	39	0.00	0.00%	25.6/sec	7.61	9.36	304.0
프로젝트의 멤버 조회	2	439	0	694	255.00	0.00%	1.6/sec	24.66	0.57	15742.5
특정 이슈 데이터 조회	1	145	0	145	0.00	0.00%	6.9/sec	45.10	2.44	6697.0
칸반보드 데이터 조회	1	91	0	91	0.00	0.00%	11.0/sec	65.77	4.10	6129.0
릴리즈노트 조회	1	62	0	62	0.00	0.00%	16.1/sec	51.74	5.89	3285.0
트리 데이터 조회	1	63	0	63	0.00	0.00%	15.9/sec	7.80	5.74	503.0
해결한 프로젝트 데이터 조회	1	81	0	81	0.00	0.00%	12.3/sec	39.61	4.51	3285.0
TOTAL	10	223	0	793	263.94	0.00%	4.4/sec	35.43	1.61	8156.4

시나리오 2-1. 분석 및 해석

- 모든 이슈 조회: 평균 응답 시간이 793ms 로 상대적으로 높게 나타났다. 처리량이 낮아 서비스의 부하를 견딜 수 있는지 평가해야 한다.
- 알림 내역 조회: 응답 시간이 84ms 로 빠른 응답을 보인다. 처리량과 데이터 처리량도 높은 편이다.
- 프로젝트의 릴리즈노트 조회: 평균 응답 시간이 39ms 로 빠른 응답을 보이며, 해당 기능의 가벼운 작업임을 나타낸다.
- 프로젝트의 멤버 조회: 평균 응답 시간이 439ms 로 높게 나타났다. 편차가 크며, 더 자세한 분석이 필요하다.
- 특정 이슈 데이터 조회: 평균 응답 시간이 145ms 로 중간 정도의 응답 시간을 나타내며, 처리량도 적절한 수준이다.
- 칸반보드 데이터 조회: 평균 응답 시간이 91ms 로 빠른 응답을 보이며, 처리량과 데이터 처리량도 상당히 높은 편이다.
- 릴리즈노트 조회: 평균 응답 시간이 62ms 로 빠른 응답을 보이며, 평균 전송 데이터 크기가 다소 크게 나타났다.
- 트리 데이터 조회: 평균 응답 시간이 63ms 로 빠른 응답을 보이며, 평균 전송 데이터 크기도 적당하다.
- 해결한 프로젝트 데이터 조회: 평균 응답 시간이 81ms 로 적절한 수준의 성능을 나타내며, 처리량도 적절한 편이다.

시나리오 2-2. 성능 개선 사항 및 결론

- "모든 이슈 조회" API 에서 높은 응답 시간을 보이므로, 해당 기능의 처리 과정 및 서버 성능을 분석하고 최적화 작업을 진행할 필요가 있다.
- "프로젝트의 멤버 조회" API 에서는 평균 응답 시간과 편차가 높은 편이다. 성능 이슈의 원인을 파악하고 성능 개선을 위한 대책이 필요하다.

시나리오 2-3. 결과 활용 및 추후 계획

- 결과에 따라 "모든 이슈 조회"와 "프로젝트의 멤버 조회" 시나리오에 대한 성능 개선 작업을 우선적으로 진행할 필요가 있다.

- 다른 시나리오에 대해서도 성능 변화를 모니터링하며, 필요한 경우 성능 개선 작업을 고려할 예정이다.

시나리오 2-4. 추후 계획

- "모든 이슈 조회" 시나리오의 성능 개선을 위해 데이터 처리 과정 및 서버 성능을 분석하고 최적화 작업을 진행할 계획이다.

- "프로젝트의 멤버 조회" 시나리오의 성능 이슈에 대한 원인 분석을 수행하고, 최적화 방안을 모색할 예정이다.

- 나머지 시나리오에 대해서도 성능 변화를 모니터링하고, 필요한 경우 성능 개선 작업을 진행할 계획이다.

2. 비기능 테스트: 가상 사용자별 응답시간 측정 (100, 500, 1000 명)

- 동시 접속자: 100 명

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
모든 이슈 조회	100	327	0	1047	208.15	0.00%	56.9/sec	365.14	20.15	6577.0
알림 내역 조회	100	278	0	1147	203.99	0.00%	45.9/sec	1078.38	16.88	24084.0
프로젝트의 릴리즈노트 조회	100	264	0	867	182.55	0.00%	39.0/sec	11.57	14.23	304.0
프로젝트의 멤버 조회	200	250	0	899	178.48	0.00%	54.5/sec	838.23	19.52	15742.0
특정 이슈 데이터 조회	100	401	0	1171	213.59	0.00%	29.6/sec	190.42	10.51	6577.0
칸반보드 데이터 조회	100	285	0	729	148.82	0.00%	29.7/sec	174.34	11.08	6009.0
릴리즈노트 조회	100	288	0	933	182.82	0.00%	29.6/sec	115.51	10.81	3998.0
트리 데이터 조회	100	253	0	1230	225.02	0.00%	33.1/sec	16.24	11.94	503.0
해결한 프로젝트 데이터 조회	100	180	0	638	120.13	0.00%	44.2/sec	172.53	16.14	3998.0
TOTAL	1000	278	0	1230	194.19	0.00%	260.6/sec	2126.04	94.42	8353.4

- 동시 접속자: 500 명

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
모든 이슈 조회	500	1121	0	4894	800.89	0.00%	97.8/sec	628.34	34.68	6577.0
알림 내역 조회	500	1336	0	6088	856.10	0.00%	58.9/sec	1385.78	21.69	24084.0
프로젝트의 릴리즈노트 조회	500	1201	0	5389	760.58	0.00%	43.5/sec	12.92	15.90	304.0
프로젝트의 멤버 조회	1000	1078	0	6159	818.35	0.00%	75.8/sec	1164.71	27.12	15742.0
특정 이슈 데이터 조회	500	1415	0	5901	909.79	0.00%	41.0/sec	263.62	14.55	6577.0
칸반보드 데이터 조회	500	1345	0	4786	821.66	0.00%	40.7/sec	238.85	15.18	6009.0
릴리즈노트 조회	500	1197	0	4945	788.39	0.00%	50.9/sec	188.71	18.59	3998.0
트리 데이터 조회	500	1060	0	5961	739.18	0.00%	53.3/sec	26.18	19.26	503.0
해결한 프로젝트 데이터 조회	500	668	0	3695	528.14	0.00%	56.0/sec	218.63	20.45	3998.0
TOTAL	5000	1150	0	6159	825.58	0.00%	364.9/sec	2976.36	132.19	8353.4

- 동시 접속자: 1000 명

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
모든 이슈 조회	1000	2285	0	6632	1435.63	0.00%	134.7/sec	856.87	47.76	6513.2
알림 내역 조회	1000	2710	0	6477	1028.71	0.00%	98.9/sec	2238.72	36.41	23179.0
프로젝트의 릴리즈노트 조회	1000	2058	0	7438	850.54	0.00%	73.5/sec	21.83	26.86	304.0
프로젝트의 멤버 조회	2000	2129	0	8233	970.89	0.00%	78.6/sec	1208.81	28.14	15742.0
특정 이슈 데이터 조회	1000	2840	0	9147	1112.50	0.00%	58.2/sec	314.17	21.00	5431.1
칸반보드 데이터 조회	1000	2807	0	8022	1000.93	0.00%	54.2/sec	290.56	20.23	5486.6
릴리즈노트 조회	1000	2544	0	7002	872.85	0.00%	52.2/sec	167.59	19.08	3285.0
트리 데이터 조회	1000	2238	0	6133	776.05	0.00%	55.9/sec	27.47	20.21	503.0
해결한 프로젝트 데이터 조회	1000	1511	0	5585	812.61	0.00%	71.6/sec	229.54	26.13	3285.0
TOTAL	10000	2323	0	9147	1072.71	0.00%	362.4/sec	2812.71	131.31	7947.1

1) 100 명 vs 500 명

API 1: 모든 이슈 조회

- 샘플 수: 100vs 500
- 평균 응답 시간: 606ms vs 1761ms
- 최소 응답 시간: 0ms
- 최대 응답 시간: 2196ms vs 9279ms
- 응답 시간 변동성: 512.20ms vs 1222.96ms
- 처리량: 0.08782 requests/sec vs 50.72023 requests/sec
- 데이터 처리량: 0.56 KB/sec (수신) vs 268.61 KB/sec (수신)
- 평균 전송 데이터 크기: 6564.3 bytes

분석 및 해석:

모든 이슈 조회 시나리오에서 응답 시간이 상당히 증가하였다. 평균 응답 시간은 대략 3 배 가량 늘어난 것으로 나타나며, 최대 응답 시간 역시 크게 증가했다. 응답 시간 변동성과 처리량도 크게 차이가 난다. 시스템의 응답성이 저하되었음을 나타내므로 이 부분에 대한 최적화가 필요하다.

API 2: 알림 내역 조회

- 샘플 수: 100 vs 500
- 평균 응답 시간: 412ms vs 1849ms
- 최소 응답 시간: 0ms
- 최대 응답 시간: 1450ms vs 9004ms
- 응답 시간 변동성: 310.93ms vs 1177.50ms
- 처리량: 0.08783 requests/sec vs 43.32005 requests/sec
- 데이터 처리량: 1.99 KB/sec (수신) vs 980.58 KB/sec (수신)
- 평균 전송 데이터 크기: 23179 bytes

분석 및 해석:

알림 내역 조회 API 에서도 응답 시간이 상당히 증가하였다. 평균 응답 시간은 대략 4 배 가량 늘어난 것으로 나타나며, 최대 응답 시간 역시 크게 증가했다. 응답 시간 변동성과 처리량도 크게 차이가 나며, 데이터 처리량 또한 상당히 높게 나타난다. 이 또한 시스템 성능 저하의 원인이므로 개선이 필요하다.

(이하 시나리오별 분석 생략)

두 번째 결과에서는 모든 시나리오에서 응답 시간이 크게 증가하였고, 처리량과 데이터 처리량도 매우 높게 나타났다. 이는 시스템의 성능 저하를 나타내므로, 성능 향상을 위한 최적화 작업이 필요한 상황임을 알 수 있다.

2) 500 명 vs 1000 명

API 1: 모든 이슈 조회

- 샘플 수: 500 vs 1000
- 평균 응답 시간: 1761ms vs 2265ms
- 최소 응답 시간: 0ms
- 최대 응답 시간: 9279ms vs 6632ms
- 응답 시간 변동성: 1222.96ms vs 1435.63ms
- 처리량: 50.72023 requests/sec vs 134.71642 requests/sec
- 데이터 처리량: 268.61 KB/sec (수신) vs 856.87 KB/sec (수신)
- 평균 전송 데이터 크기: 5423.0 bytes

API 2: 알림 내역 조회

- 샘플 수: 500 vs 1000
- 평균 응답 시간: 1849ms vs 2710ms
- 최소 응답 시간: 0ms

- 최대 응답 시간: 9004ms vs 6477ms
- 응답 시간 변동성: 1177.50ms vs 1028.71ms
- 처리량: 43.32005 requests/sec vs 98.90219 requests/sec
- 데이터 처리량: 980.58 KB/sec (수신) vs 2238.72 KB/sec (수신)
- 평균 전송 데이터 크기: 23179 bytes

(이하 시나리오별 분석 생략)

분석 및 해석:

두 번째 결과에서는 모든 API 에서 응답 시간이 증가하였고, 처리량과 데이터 처리량도 상승한 것으로 나타난다. 시스템의 성능이 저하되었으며, 응답 시간 변동성도 증가한 것을 확인할 수 있다. 이러한 상황에서는 성능 향상을 위한 최적화 작업이 필요하다. 또한, 데이터 처리량과 관련하여 수신 데이터의 증가도 관찰된다.

모든 API 에서 응답 시간이 증가하였으며, 특히 알림 내역 조회 시나리오에서는 최대 응답 시간과 응답 시간 변동성이 감소하였다. 그러나 처리량과 데이터 처리량이 크게 증가하였으며, 이로 인해 시스템의 부하가 증가하고 있음을 알 수 있다.

총 처리량과 데이터 처리량이 두 번째 결과에서 상당히 증가하였다. 이로 인해 시스템 성능의 저하와 부하 증가가 나타나며, 이로 인해 사용자 경험이 저하될 수 있다. 따라서 성능 향상을 위한 최적화 및 리소스 관리가 필요하다.

3) 100 vs 1000 명

API 1: 모든 이슈 조회

- 샘플 수: 101 vs 1000
- 평균 응답 시간: 606ms vs 2265ms
- 최소 응답 시간: 0ms
- 최대 응답 시간: 2196ms vs 6632ms
- 응답 시간 변동성: 512.20ms vs 1435.63ms
- 처리량: 0.08782 requests/sec vs 134.71642 requests/sec
- 데이터 처리량: 0.56 KB/sec (수신) vs 856.87 KB/sec (수신)

- 평균 전송 데이터 크기: 6564.3 bytes

API 2: 알림 내역 조회

- 샘플 수: 101 vs 1000
- 평균 응답 시간: 412ms vs 2710ms
- 최소 응답 시간: 0ms
- 최대 응답 시간: 1450ms vs 6477ms
- 응답 시간 변동성: 310.93ms vs 1028.71ms
- 처리량: 0.08783 requests/sec vs 98.90219 requests/sec
- 데이터 처리량: 1.99 KB/sec (수신) vs 2238.72 KB/sec (수신)
- 평균 전송 데이터 크기: 23179 bytes

(이하 시나리오별 분석 생략)

분석 및 해석:

두 번째 결과에서는 모든 시나리오에서 응답 시간이 상당히 증가하였다. 특히 알림 내역 조회 시나리오에서는 평균 응답 시간과 최대 응답 시간이 매우 높아지면서 응답 시간 변동성이 줄어들었다. 또한, 처리량과 데이터 처리량이 상당히 증가하여 시스템의 부하가 높아졌다.

총 처리량과 데이터 처리량 역시 두 번째 결과에서 상당히 증가하였다. 이로 인해 시스템의 성능 저하와 부하 증가가 나타나며, 사용자 경험이 저하될 수 있다. 이를 고려하여 시스템 최적화와 리소스 관리가 필요하다.

결론

Releasy 웹 애플리케이션은 대체로 안정적인 성능을 보여주었으나, 일부 시나리오에서 응답 시간이 예상보다 길어지는 문제가 확인되었다. 이를 해결하기 위해 데이터베이스 성능 최적화와 코드 리팩토링을 할 예정이다.

추후 계획

- DB 처리량 최적화(스레드 풀 및 DB 커넥션 풀 크기 조정 또는 DB 서버 스케일 아웃) 작업 진행
- DB 처리 시간 최적화(쿼리 튜닝, 캐싱)
- 대기 시간 최적화 (정적 파일 트래픽 분리: CDN 사용)
- 추가 성능 테스트 진행 후 개선 여부 평가

이 하 여 백